AD-A047 254 MITRE CORP MCLEAN VA METREK DIV F/G 17/7
INITIAL COLLISION AVOIDANCE ALGORITHMS FOR THE BEACON-BASED COL--ETC(U)
APR 77 J CLARK, A MCFARLAND DOT-FA69NS-162
UNCLASSIFIED MTR-7532 FAA-RD-77-163 NL

1 OF 2

AD A047254

AD A047254

(18) (19)

Report No. FAA-RD-77-163

12 mc

(6)

Initial Collision Avoidance Algorithms for the Beacon-based Collision Avoidance System.

(10)

J. CLARK
A. MCFARLAND

(14) MTR-7532

(11) APRIL 1977

(12) 96 p.

(15) DOT-FA69NS-162

Document is available to the U.S. public through
the National Technical Information Service,
Springfield, Virginia 22161.

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D.C. 20590

409 890

mt

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FAA-RD-77-163		

4. Title and Subtitle
Initial Collision Avoidance Algorithms for
The Beacon-based Collision Avoidance System

5. Report Date
April 1977

6. Performing Organization Code
W-46

7. Author(s)
J. Clark, A. McFarland

8. Performing Organization Report No.
MTR-7532

9. Performing Organization Name and Address
The MITRE Corporation
METREK Division
1820 Dolly Madison Blvd
McLean, Virginia 22101

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DOT-FA69NS-162

12. Sponsoring Agency Name and Address
Department of Transportation
Federal Aviation Administration
Systems Research and Development Service
Washington, D. C. 20591

13. Type of Report and Period Covered
Project Report

14. Sponsoring Agency Code
ARD-250

15. Supplementary Notes

16. Abstract

This document describes a set of baseline collision avoidance algorithms which can be used as a point of departure for the development of final algorithms for the FAA's Beacon-based Collision Avoidance System (BCAS). The algorithms were structured to permit great flexibility in an experimental environment such as NAFEC. They incorporate a number of selectable options in the collision avoidance logic and in the display output. One option permits the selection of either a passive mode logic or an active mode logic. When the passive mode is selected, other options allow horizontal positive or negative commands to be used. In addition, the display of positive or negative commands can be selected or suppressed, and limit vertical rate commands can be selected for display independently of positive or negative commands. Two types of Intruder Position Data (IPD)-flashing IPD's and ordinary IPD's-can also be selected for display. The logic can drive three types of cockpit displays namely, an ACAS display, a baseline IPC display, and a general purpose Plan View Display (PVD).

17. Key Words	18. Distribution Statement
BCAS, Threat logic, Collision Avoidance Algorithms, Commands	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	95	

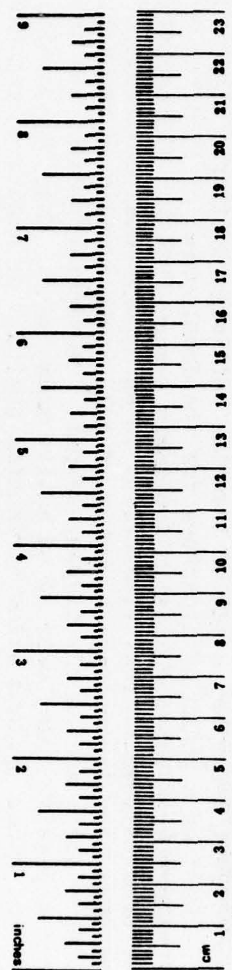
Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

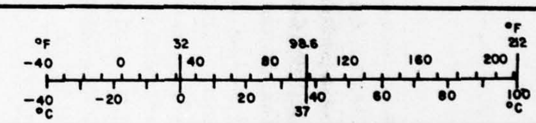


TABLE OF CONTENTS

TABLE OF CONTENTS
(CONCLUDED)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(CONCLUDED)

1. INTRODUCTION

This document describes a set of baseline collision avoidance algorithms that can be used as a point of departure for the development of final algorithms for the Federal Aviation Administration's (FAA) Beacon-based Collision Avoidance System (BCAS). The algorithms have been structured for great flexibility in an experimental environment such as the FAA's National Aviation Facility Experimental Center (NAFEC). They incorporate a number of selectable options in the collision avoidance logic and in the display output. As a result, they can represent logic for the active mode or the passive mode of a BCAS system. No logic has been defined specifically for the semi-active mode because such a logic would not be significantly different from the logic for the passive mode.

Options can be selected to use horizontal positive or negative commands with the passive mode logic, and options can be selected for the display of several types of information to the pilot. The display of positive or negative commands can be selected or suppressed. Limit vertical rate commands can be selected for display independently of the selection of positive or negative commands. Two types of Intruder Position Data (IPD)--flashing IPD's and ordinary IPD's--can be selected for display.

In addition, the logic presented here can drive three types of cockpit displays. The first is the Airborne Collision Avoidance System (ACAS) display which is used with the logic contained in Reference 1. The second is the baseline Intermittent Positive Control (IPC) display described in Reference 2. The third display is a general purpose Plan View Display (PVD) which can present a pictorial plan view of intruder aircraft in the vicinity of the protected aircraft in an own-aircraft-centered and own-heading--oriented framework.

The algorithms in this document have been structured specifically to interface with the Digital Simulation Facility (DSF) real-time air traffic control simulation vehicle at NAFEC. The algorithms could rather readily be adapted for other real-time or fast-time simulation environments. However, some major revision of the external interfaces and some additions to the logic itself would be required to interface this collision avoidance logic with the remainder of the BCAS system in a live flight environment.

The logics described here are only skeletal logics in several respects. While the logic is capable of displaying several IPD's simultaneously, no provision has been made to treat

situations in which more than one intruder is a sufficient threat to require positive or negative commands. No logic has been specified to define how the logic and display output should change as the surveillance mode of a tri-modal BCAS changes from one mode to another. A number of other similar details have been ignored or simplified in defining the present version of the system. It is intended that these items be addressed as the development of the BCAS collision avoidance algorithms proceeds. Over time, a number of the selectable options should disappear and the collision avoidance logic should converge to an accepted mainline design.

The parameter values presented in Appendix B are intended only to indicate the order of magnitude of final parameters. A firm recommendation of a set of parameters for use in actual simulation will be provided prior to production testing.

2. HIGH LEVEL LOGIC AND EXTERNAL INTERFACES

This section describes the data structures assumed in writing this document, presents the overall flow of the logic, presents the assumed structure of the external simulation program, and presents the detailed interface between the simulation program and the BCAS collision avoidance logic.

2.1 BCAS Collision Avoidance Logic Internal Data Structures

There are three major data items assumed in writing this document--the own aircraft state vector, the intruder state vector and the display vector. In addition, decision tables and parameter lists are presented in later sections of this document.

The own aircraft state vector, presented in Table 2-1, contains all data pertaining to own aircraft which must be retained for some period of time. Local variables not used outside a single subroutine are not a part of the own aircraft state vector. This same state vector is used for both the active mode and the passive mode simulations. The X and Y position and velocity fields are not used with the active mode logic.

The intruder state vector, presented in Table 2-2, contains all data, from own aircraft's point of view, that is unique to an individual intruder. This state vector is also used with both the active mode and the passive mode of operation. Intruder state vectors for all intruders of concern to own aircraft are linked together into a list. The first intruder state vector is pointed to by the FSTINT pointer in own aircraft's state vector.

The display vector is presented and described in detail in Section 6. A common display vector is used to drive any of the displays that might be used with the BCAS logic. The display vector contains all of the data pertaining to a single intruder that would be needed to drive a display. Data for several intruders can be displayed simultaneously to own aircraft. In this case, a separate display vector will exist for each intruder and all will be linked together in a list with the first display vector pointed to by FSTVEC.

Own aircraft maintains an intruder state vector on all aircraft that are potential threats to own aircraft. The criteria for maintaining an intruder state vector are considerably broader than the criteria for issuing commands or displays of intruder position data. In this way, sufficient time to establish tracking on an intruder will be available before commands or IPD data must be displayed for that intruder. Hence, there will not

TABLE 2-1

OWN AIRCRAFT STATE VECTOR

SYMBOL	UTILIZATION
IDOWN	Identification of own aircraft
XOWN YOWN ZOWN	Own aircraft tracked position coordinates
XDOWN YDOWN ZDOWN	Own aircraft tracked velocity coordinates
TDATA	Time for which the tracked position and velocity coordinates are represented
SLEVEL	Desensitization level indicator
RTRANS	Range of own aircraft from the closest ground transponder used for desensitization
INDEX	Index corresponding to desensitization level used for entering the 3 x 2 matrix of values for the settable parameters
NTENT	Own aircraft intent code
TPROV	Time at which NTENT is set to provisional status
TPOSIT	Time at which DPLY is set to a positive command
CONINT	Identification of the intruder controlling the display of commands
FSTINT	Pointer to the first intruder state vector in the list of intruder state vectors for own aircraft
FSTVEC	Pointer to the first display vector in the list of display vectors for own aircraft

TABLE 2-2

INTRUDER STATE VECTOR

SYMBOL	UTILIZATION
IDINT	Identification of the intruder
XINT YINT ZINT	Intruder's tracked position coordinates
XDINT YDINT ZDINT	Intruder's tracked velocity coordinates
R	Tracked range to the intruder
RD	Tracked range rate of the intruder
TDATA	Time for which the tracked position and velocity coordinates are represented
TREPT	Time of the last set of target reports for this intruder
EQ	Intruder's equipage with BCAS: 0 if unequipped, 1 if equipped
MTENT	Intruder's intent code
KHIT	Own intent status indicator with respect to this intruder
JNDEX	Index corresponding to intruder's equipage used for entering the 3x2 matrix of values for the settable parameters
FLASH	Flag indicating whether own aircraft has a flashing IPD for this intruder
CMDSAV	Command being displayed to own aircraft due to this intruder (Coding is the same as DPLY but value may be different from DPLY)
NXTINT	Pointer to the next intruder state vector in own aircraft's list of intruder state vectors

be a one-for-one correspondence between display vectors and intruder state vectors.

The relationships between the data structures are shown pictorially in Figure 2-1. The simulation program sets aside arrays of the three data vectors of sufficient size to cover the expected traffic scenario. For each aircraft active in the simulation scenario, an own aircraft state vector is activated and initialized. When a new intruder becomes a potential threat to a given aircraft, an intruder state vector is activated and linked to the list of intruders for that aircraft. When a display message is generated for own aircraft by a given intruder, a new display vector is activated and linked to own aircraft's list of display vectors.

Figure 2-1 shows that the aircraft having the fourth own aircraft state vector has three intruders on its list of intruders and has two display vectors on its list of display vectors.

2.2 High Level Organization of the BCAS Logic

The highest level of flow through the BCAS logic is best explained by presenting the assumed structure of the external simulation program. The overall flow chart of the simulation operation is shown in Figure 2-2.

The BCAS programs may or may not be executed on every simulation cycle. When the active mode logic is under test, it is recommended that the BCAS program be executed every second. For the passive mode logic, it is recommended that the BCAS program be executed every two seconds to simulate the effective data rate of the passive mode.

There are two types of calls made to the BCAS programs. One is a call (to TROACT/TROPAS), once per BCAS logic cycle* for each BCAS aircraft for the purpose of conducting tracking on own-aircraft coordinates. This is done regardless of whether or not a particular aircraft is in a conflict situation. Other house-keeping tasks which need to be done on a periodic basis will also be done in this call. The second call (to BCASDT), is one in which own aircraft receives a report of data relating to a specific intruder in potential conflict with own aircraft.

* This phrase is used throughout the document. All of the logic flow in Figure 2-2 from the YES branch of the diamond "TIME TO CALL BCAS LOGIC?" to the diamond "END OF SIMULATION?" constitutes one BCAS logic cycle.

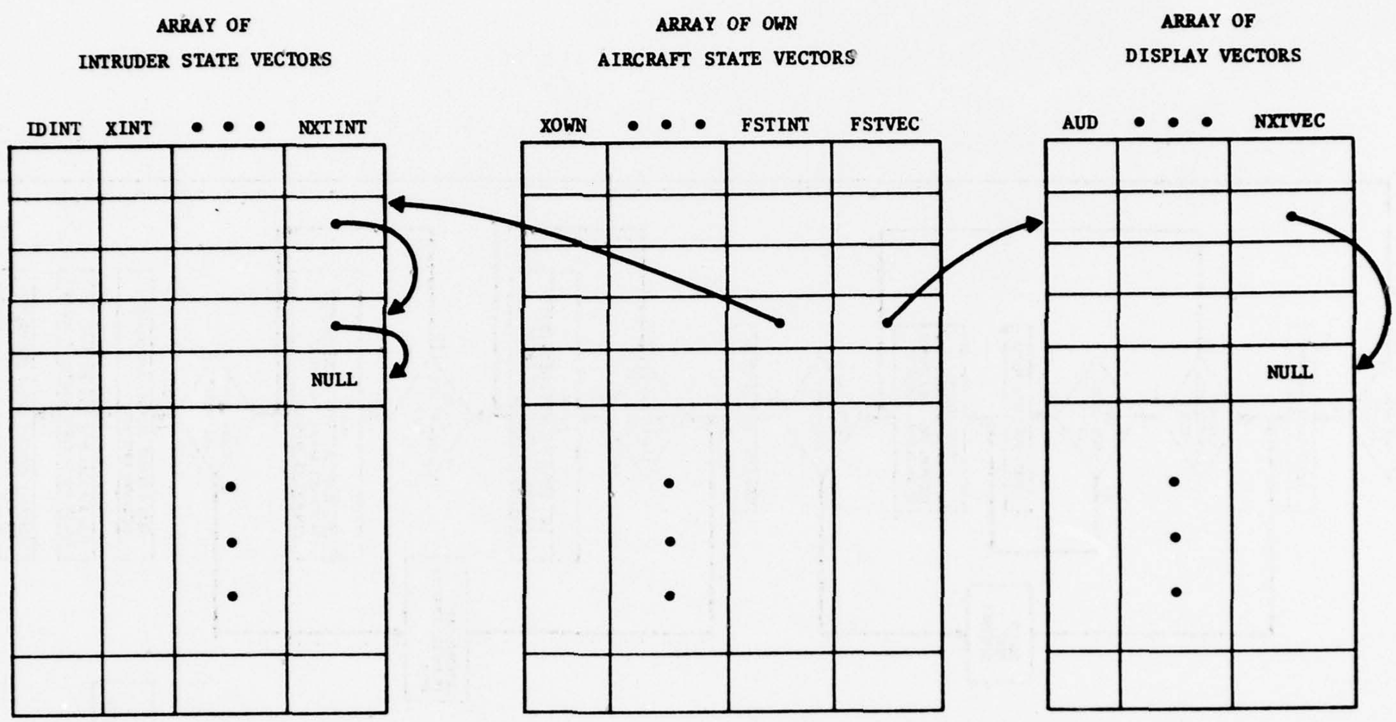


FIGURE 2-1
GRAPHICAL DEPICTION OF THE BCAS DATA STRUCTURES

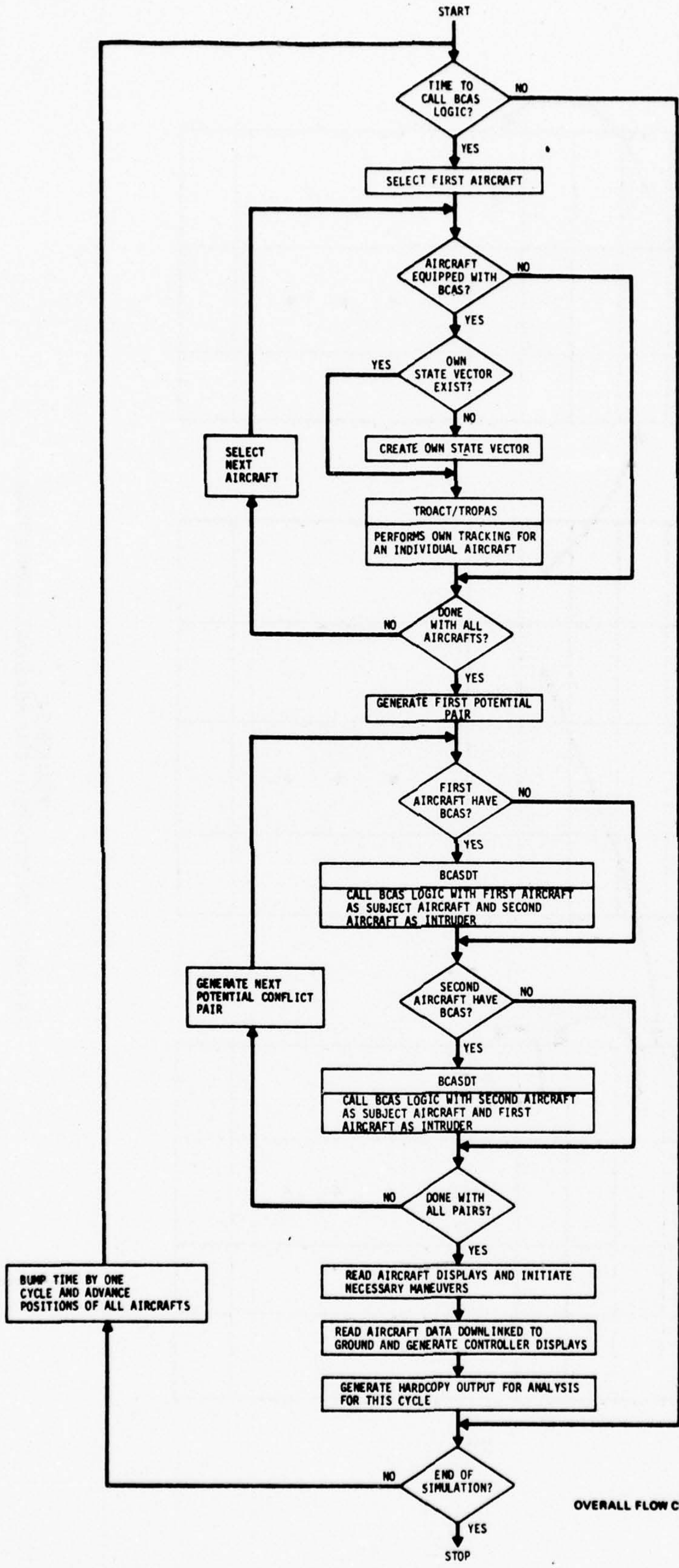


FIGURE 2-2
OVERALL FLOW CHART OF SIMULATION PROGRAM OPERATIONS

At several places in this document, two sets of flow charts are presented for the same subroutine--one is for use with the active mode logic and one is for use with the passive mode logic. It is left to the discretion of those coding the BCAS logic for simulation as to whether the subroutines should be coded separately and one selected at program load time, or whether they should be coded together and selected with an option switch set at execution time, or whether a union of the code from the two sets of flow charts should be coded as a single subroutine with enbedded tests to bypass unnecessary logic.

From the point of view of this document, it is the responsibility of the external simulation program to activate an own aircraft state vector when a new aircraft enters the simulation scenario.

The subroutines TROACT and TROPAS are presented in Section 7. The parameters in the subroutine calls represent one of the external interfaces of the BCAS logic and are also presented in Section 7. These routines are written to provide tracking on only a single aircraft.

The external simulation program is assumed to be able to identify pairs of potentially conflicting aircraft through some coarse screening process. Once a potentially conflicting pair has been identified, each aircraft (if BCAS equipped) of the pair is treated as own aircraft and the other aircraft is presented to it as an intruder. Coarse screen criteria of three nautical miles or 75 second tau for the horizontal and 3300 feet or 75 second tau for the vertical should provide adequate tracking time prior to alerts.

The subroutine BCASDT represents all of the logic that would be performed by the BCAS collision avoidance logic in a real-world BCAS system upon receiving a report from an intruder that had been determined to be in potential conflict with own aircraft. A high level flow chart of this subroutine is presented in Figure 2-3. Only the high level flow is presented here. Individual tasks are represented as subroutines and are described individually in other sections of this document. The numbers in parentheses with the subroutine names give the major section numbers describing the subroutines.

The BCASDT subroutine performs intruder tracking, threat detection and resolution, limit command determination, IPD determination and display functions. The program tests various option switches to determine which command and display options have been selected. The selectable options are summarized in Table 2-3. Except for PAFLG, NOHOR, and NOHPOS, the option is selected if the switch is

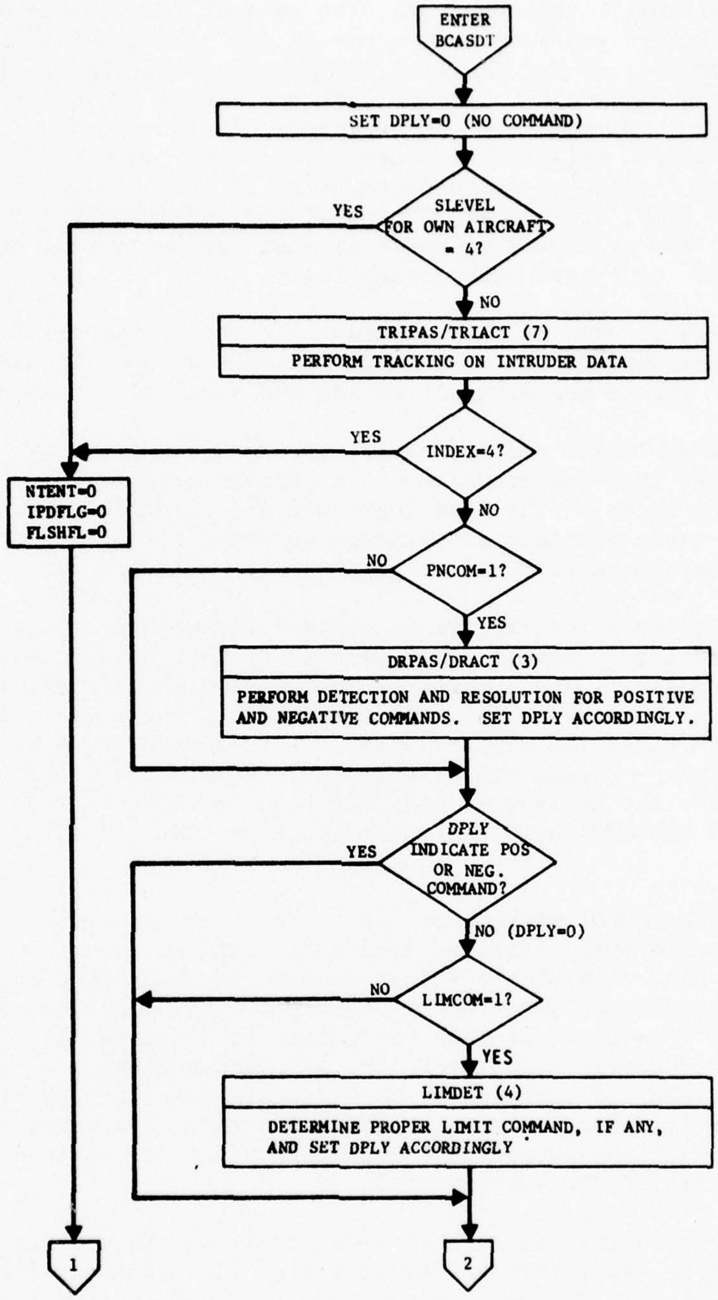


FIGURE 2-3
HIGH LEVEL FLOW CHART FOR BCASDT

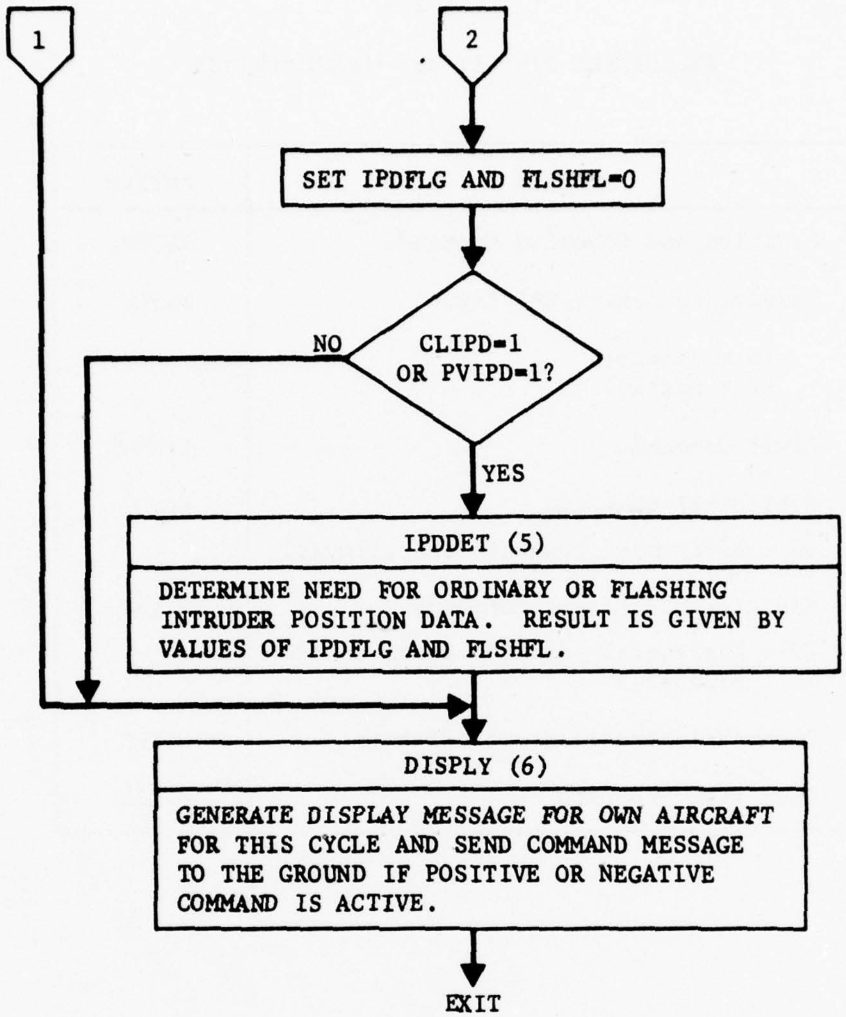


FIGURE 2-3 (Concluded)
HIGH LEVEL FLOW CHART FOR BCASDT

TABLE 2-3

SELECTABLE DISPLAY AND LOGIC OPTIONS

OPTION	SWITCH
Positive and Negative Commands	PNCOM
Passive or Active CAS Logic (0 = passive) (1 = active)	PAFLG
Limit Commands	LIMCOM
Horizontal Commands (0 = horizontal commands are allowed)	NOHOR
Positive Horizontal Commands (0 = horizontal positive commands are allowed)	NOHPOS
Clock/Relative Altitude IPD Data	CLIPD
Plan View IPD Data	PVIPD

set equal to 1 and not selected if the switch is set equal to zero. As mentioned above, more extensive use of PAFLG than appears explicitly could be made in the actual implementation of the BCAS programs. In any case, the functional capability to select either active mode or passive mode logic is present.

The program initially sets the display indicator to "no command" (DPLY=0). The first test in this subroutine is to determine if the BCAS logic has been disabled by the ground ATC system. This is indicated by SLEVEL=4. If so, some control variables are initialized and only the display subroutine is called so that any activity that might have existed on the display will be wiped out. It is envisioned that this level of desensitization would be exercised only at 1 or 2 miles from an airport. The BCAS logic would be shut off to avoid nuisance alarms from stationary aircraft on the airport or from aircraft in VFR visual patterns at the airport.

If SLEVEL is not equal to 4, the BCASDT subroutine calls either TRIPAS or TRIACT to track the intruder's data and then proceeds to determine the type of command, if any, which should be displayed. The program then tests INDEX to determine whether TRIPAS or TRIACT has disabled the BCAS logic. If INDEX≠4, the program then tests PNCOM to determine whether positive and negative commands have been selected. If so, the program then calls DRACT or DRPAS to perform the threat detection and resolution functions. The passive or active logic sets the display indicator DPLY to the desired display code. Regardless of whether or not the passive and the active mode logics have been bypassed, BCASDT tests DPLY to see if it is set to either a positive or a negative command. If so, the limit command logic is automatically bypassed, but if not, LIMCOM is tested to determine whether the limit logic should be entered or bypassed.

If the limit command logic is entered, further tests are made to determine the type of limit command, if any, that should be displayed; and the display indicator DPLY is set accordingly. Next, the program resets the control variables, IPDFLG and FLSHFL and then enters the display logic described in Section 5. Depending on the display options selected, the display routine computes the display vector which subsequently drives the desired displays.

After all potential conflict pairs have been processed through BCASDT, the simulation program reads the display data for all aircraft and initiates any aircraft maneuvers that are required in response to BCAS commands that are displayed. No detailed flow chart is presented for this task but several functions must

be performed here. For each BCAS aircraft, the simulation program should first clear the display of all existing data and then should read and display each display vector on that aircraft's list of display vectors. Multiple IPD's may result from the several display vectors. However, the subroutine DISPLY will have insured that only one of the display vectors contains a positive, negative, or limit command. Having read all display vectors, the simulation program should release all display vectors and reset FSTVEC to null. This method of driving the displays results in a complete refresh of the display each BCAS logic cycle.

As the next step, the simulation program should read the messages that have been created by the BCAS logic system for transmission to the ground ATC system and should generate the appropriate displays for the controller.

The last step is to generate the output for the current BCAS logic cycle that will be used for post-mission analysis.

2.3 BCAS Logic External Interfaces

This section summarizes all the external interfaces, many of which are described elsewhere in the document. The BCAS logic receives inputs pertaining to own aircraft in the call to TROPAS or TROACT. It receives inputs pertaining to an intruder in the call to BCASDT.

The simulation program will simulate missed reports in the report data from intruders. However, the BCAS logic expects a call from the simulation program each BCAS cycle for each intruder. When a missed report is simulated, a flag in the call to BCASDT is set to indicate a missed report.

The simulation program need not take special action when a new intruder first begins to satisfy the potential conflict criteria for a given subject aircraft. The BCAS logic will recognize, on the first call of BCASDT with that intruder, that no intruder state vector exists for that intruder and will activate an intruder state vector. Likewise, when an intruder ceases to satisfy the potential conflict criteria, the simulation program need take no special action. The BCAS logic will drop the intruder state vector in TROPAS or TROACT after a sufficient interval of time.

The primary output of the BCAS logic is, of course, the data going to own aircraft's display. The format of this output is the BCAS display vector presented in Section 6. The command message to the ground is an additional output.

At the logical level, there is an additional input and output associated with the BCAS logic. The input is the intruder's intent variable, MTENT, and the output is own intent, NTENT. The intruder's intent is made available in the call to BCASDT and the current value is stored in the intruder state vector in TRIPAS or TRIACT. When the simulation program makes the call to BCASDT, it should obtain the value to be used for MTENT in the calling statement directly from the appropriate own aircraft state vector. For instance, assume that aircraft A and B have been found to be in potential conflict by the coarse screening procedure. The simulation program first calls BCASDT with A as own aircraft and B as the intruder. It obtains the data for MTENT in this call from the NTENT field in B's own aircraft state vector. The procedure is comparable when the call to BCASDT is with B as own aircraft and A as the intruder. Thus, the act of outputting own intent data does not appear explicitly in the simulation structure, but is present in fact.

Table 2-4 summarizes the external interfaces of the BCAS logic. The variables in the calls to TROPAS and TROACT are explained in Section 7. The variables in the call to BCASDT, except for PAFLG, are a union of the variables in the calls to TRIPAS and TRIACT, which are described in Section 7. PAFLG is passed to the BCAS logic so that the proper version of the routines which are duplicated within BCASDT can be selected. If PAFLG indicates the passive mode, XRINT and YRINT will contain data but RR will be empty. For the active mode, the reverse is true.

TABLE 2-4

INPUTS AND OUTPUTS OF THE BCAS LOGIC

INPUTS

- CALL TO TROPAS

TCUR	IDOWN	XROWN	YROWN	ZROWN	SLEVEL	RTRANS

- CALL TO TROACT

TCUR	IDOWN	ZROWN	SLEVEL	RTRANS

- CALL TO BCASDT

TCUR	IDOWN	IDINT	RPTFLG	XRINT	YRINT	ZRINT	RR	EQ	MTENT	PAFLG

OUTPUTS

- OWN AIRCRAFT DISPLAY VECTOR

AUD	FLASH	COMACT	COMND	CLKACT	CLOCK	RELALT	PVACT	RANGE	BEAR	ALT	EQ	INTCOM	SCALE

- COMMAND MESSAGE TO GROUND

OWN IDENTIFICATION	POSITIVE OR NEGATIVE COMMAND

- TRANSMISSION OF OWN INTENT DATA BY SIMULATION PROGRAM

3. DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS

This section presents the threat detection and resolution logic that is used to determine positive and negative commands. Completely separate flow charts are presented for the logic that would be used by the active modes and the passive modes. The active mode logic is largely a subset of the passive mode logic. For this reason, the passive mode logic is presented first and is described in some detail. The active mode logic is then easily understood.

The passive mode logic (DRPAS) is presented in Figure 3-1. Either the passive mode or active mode logic will only be executed if the PNCOM switch has been selected. Note that DPLY is set to zero before entering the positive/negative command logic. The output of this subroutine is conveyed by DPLY. If no positive or negative commands are required, DPLY remains zero. Otherwise, DPLY represents the command to be displayed according to the coding presented in Appendix C.

3.1 Desensitization

An important characteristic of the detection and resolution logic is that most of the parameters are variable and are determined by own aircraft's location and by intruder's equippage. The process of setting the thresholds as a function of own aircraft's location is referred to as desensitization. The level of desensitization can be controlled by the ground air traffic control (ATC) system or it can be determined by the BCAS logic itself.

The setting of the desensitization level is done as a part of tracking of own data and is discussed in Section 7. The result is conveyed to the detection and resolution subroutine through the value of INDEX. Determining the equippage of the intruder is done during tracking of the intruder and the result is represented by the value of JNDEX.

When used to set detection thresholds, INDEX can take three different values and JNDEX can take two different values. For convenience of access, the six values associated with a parameter are stored in a doubly-dimensioned array which is referenced using INDEX and JNDEX. Thus, for example, the modified tau distance, DMOD, is set to DMOD(INDEX, JNDEX). Setting the values of the setable parameters is the first function performed in the detection and resolution subroutines.

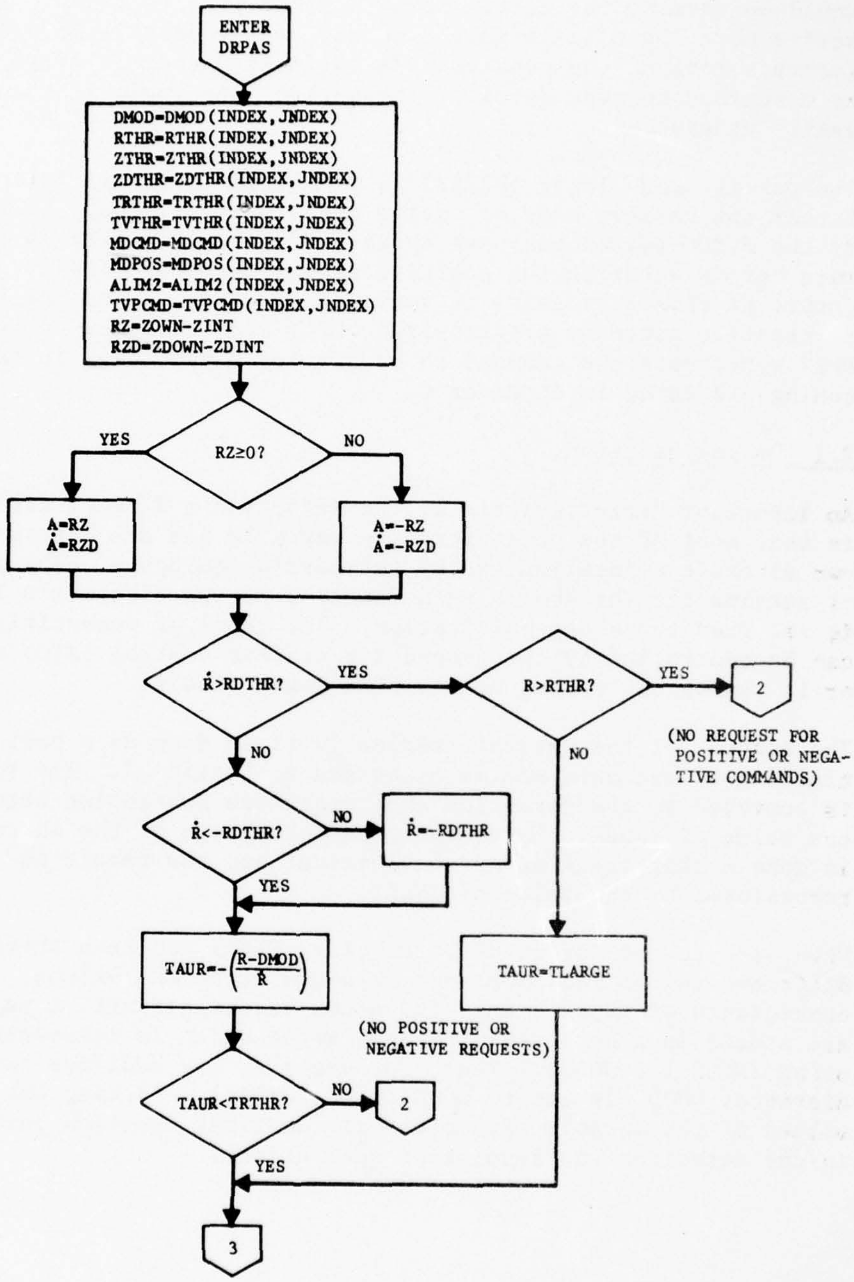


FIGURE 3-1
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS – PASSIVE MODE

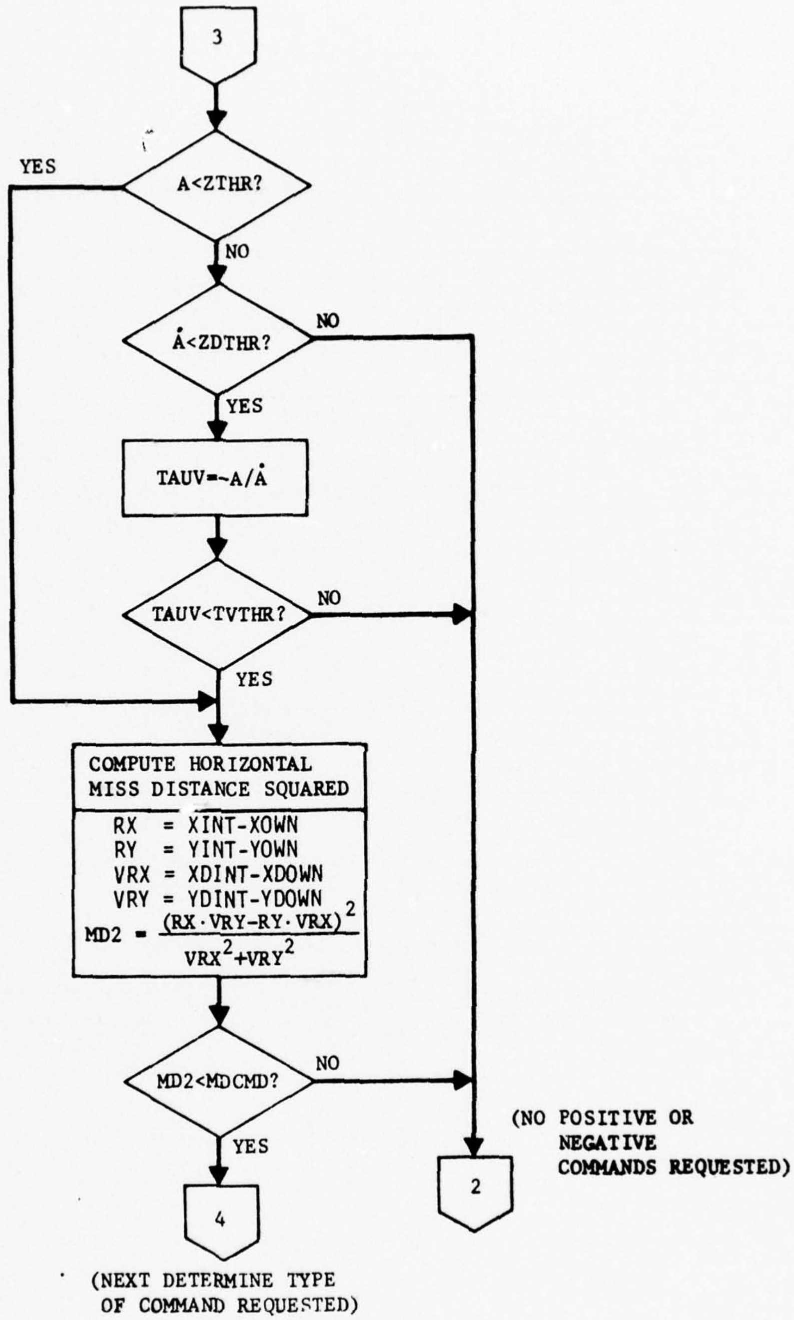


FIGURE 3-1 (Continued)
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS – PASSIVE MODE

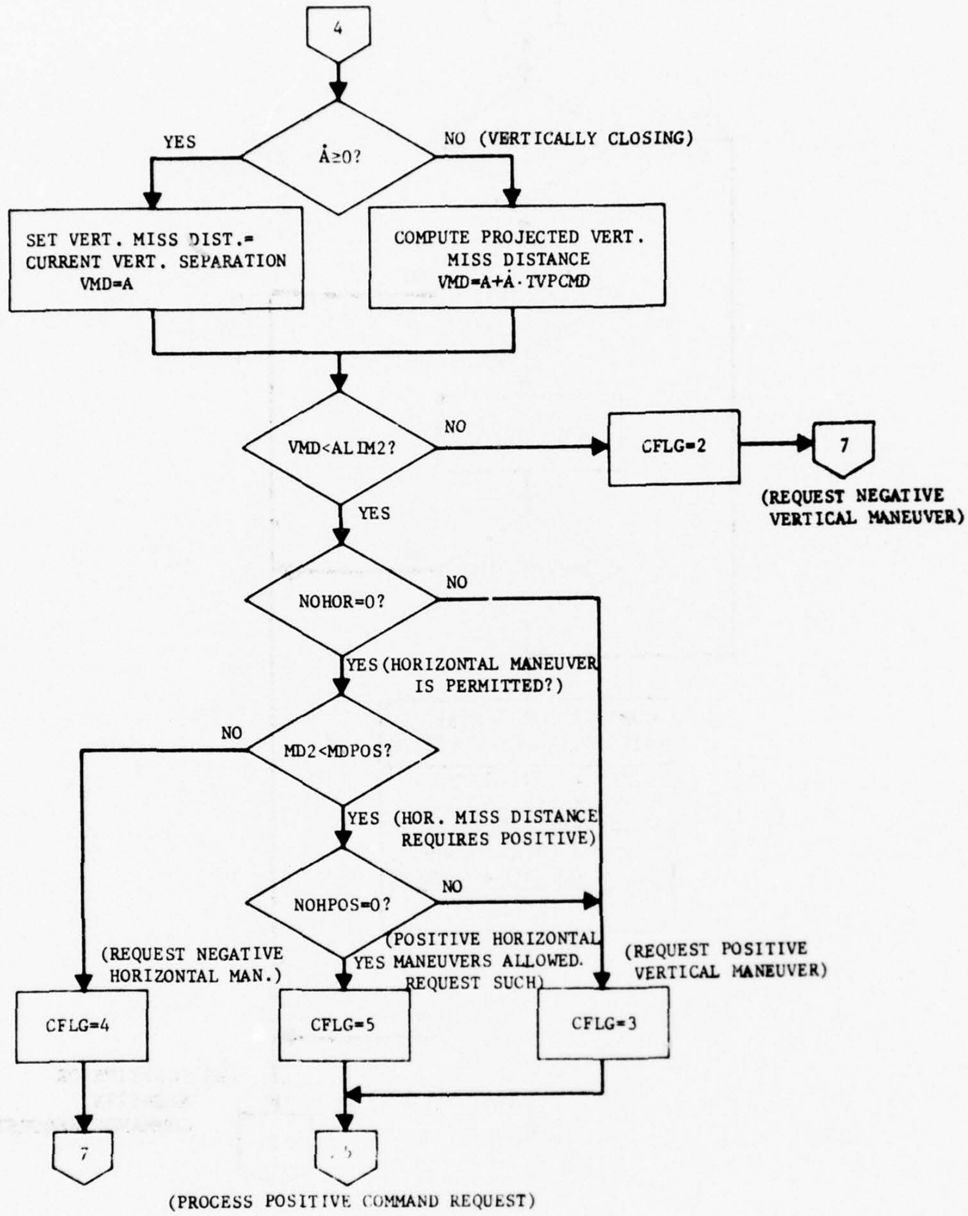


FIGURE 3-1 (Continued)
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS – PASSIVE MODE

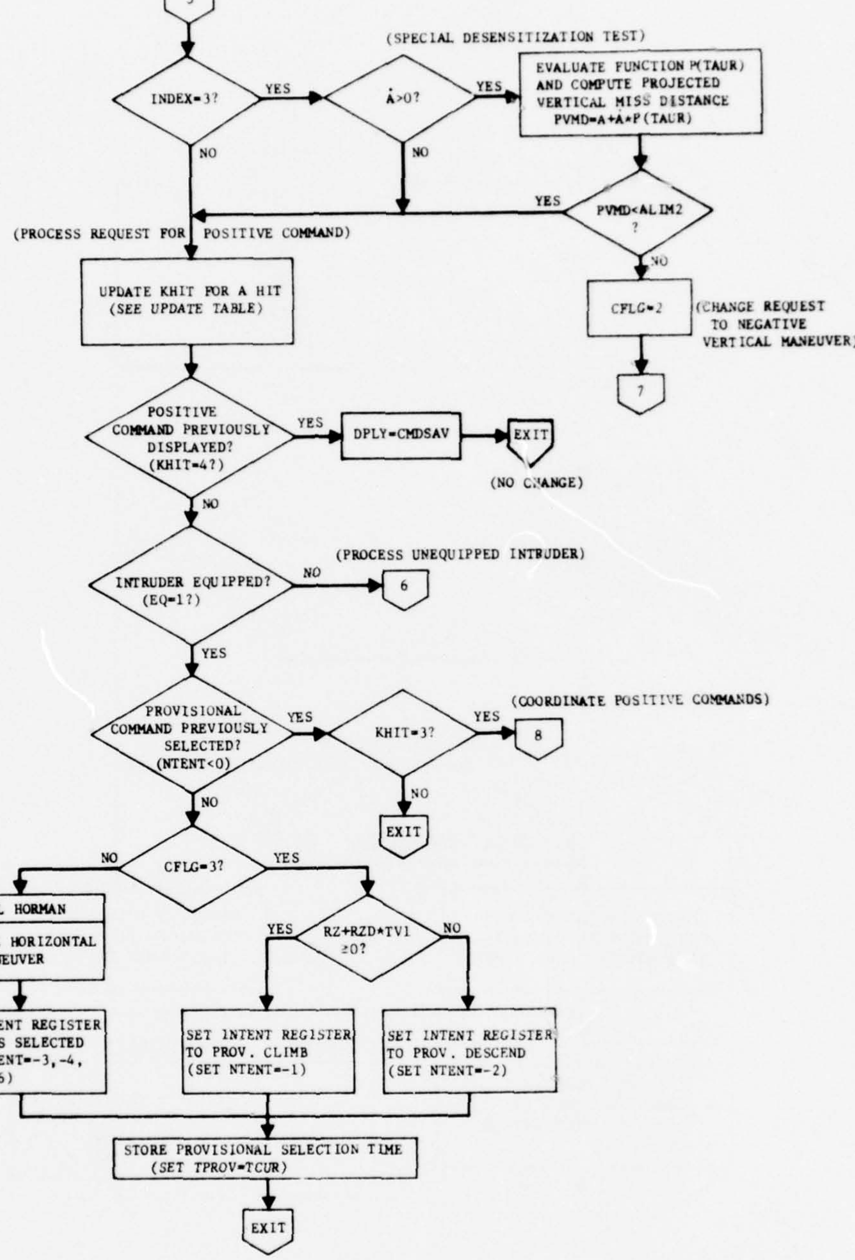
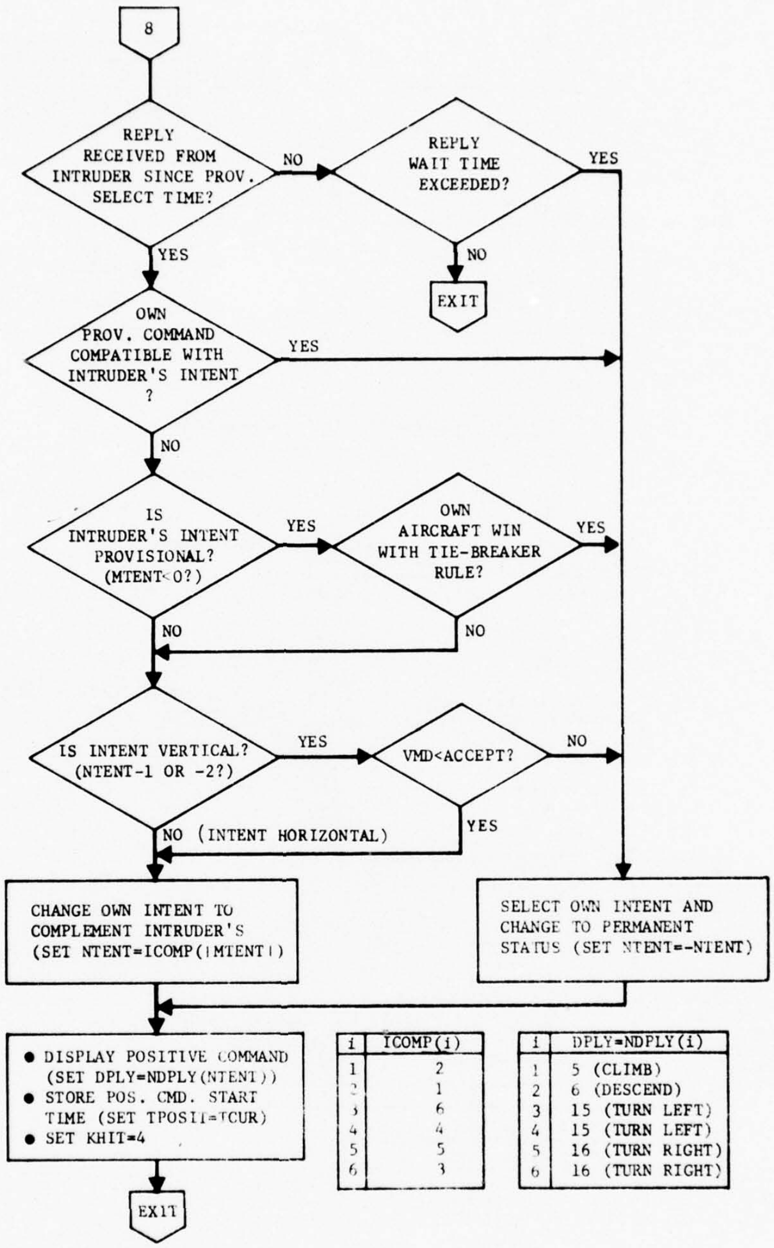


FIGURE 3-1 (Continued)
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS – PASSIVE MODE

(COORDINATE POSITIVE COMMANDS)



i	ICOMP(i)
1	2
2	1
3	6
4	4
5	5
6	3

i	DPLY=NDPLY(i)
1	5 (CLIMB)
2	6 (DESCEND)
3	15 (TURN LEFT)
4	15 (TURN LEFT)
5	16 (TURN RIGHT)
6	16 (TURN RIGHT)

FIGURE 3-1 (Continued)
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS PASSIVE MODE

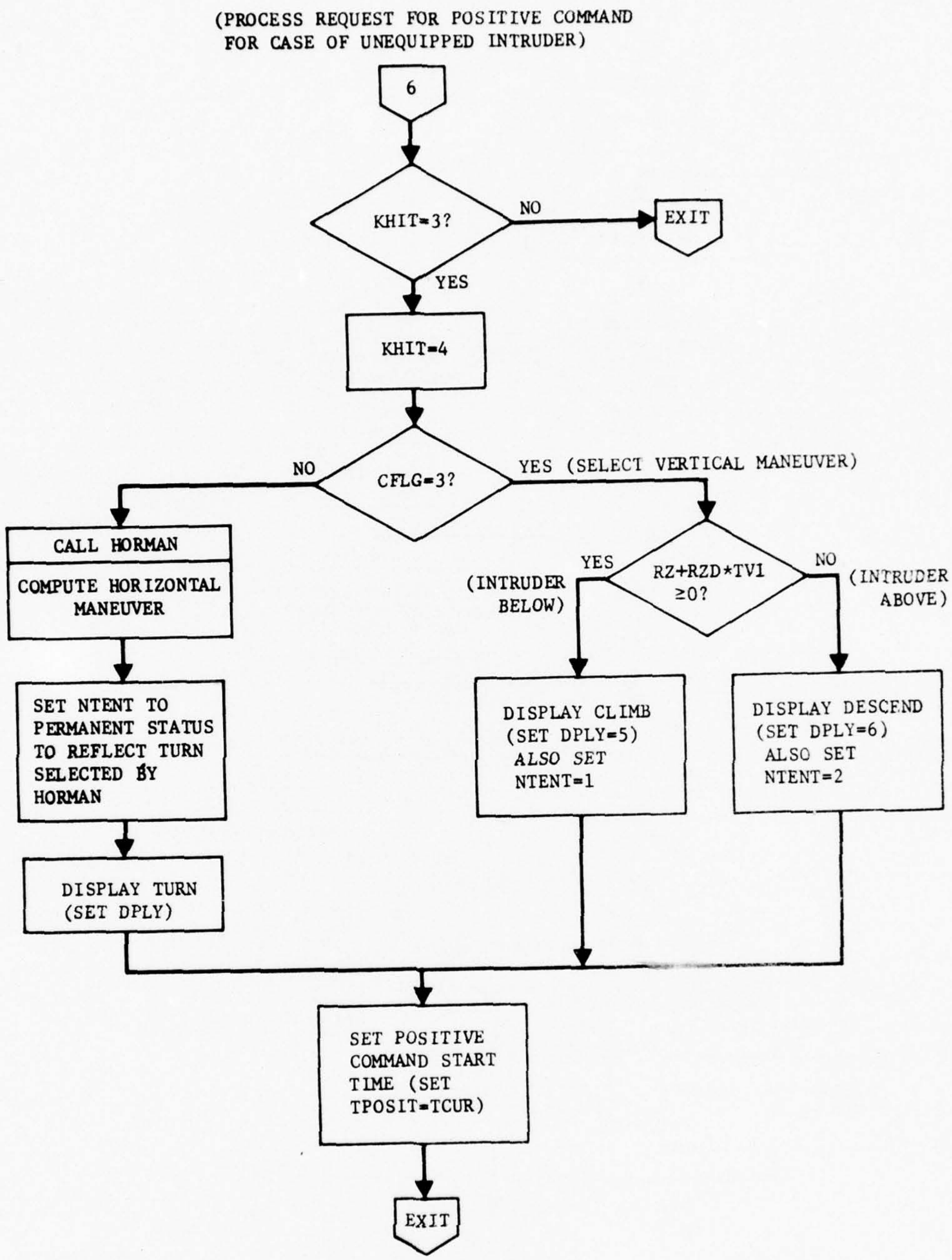


FIGURE 3-1 (Continued)
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS – PASSIVE MODE

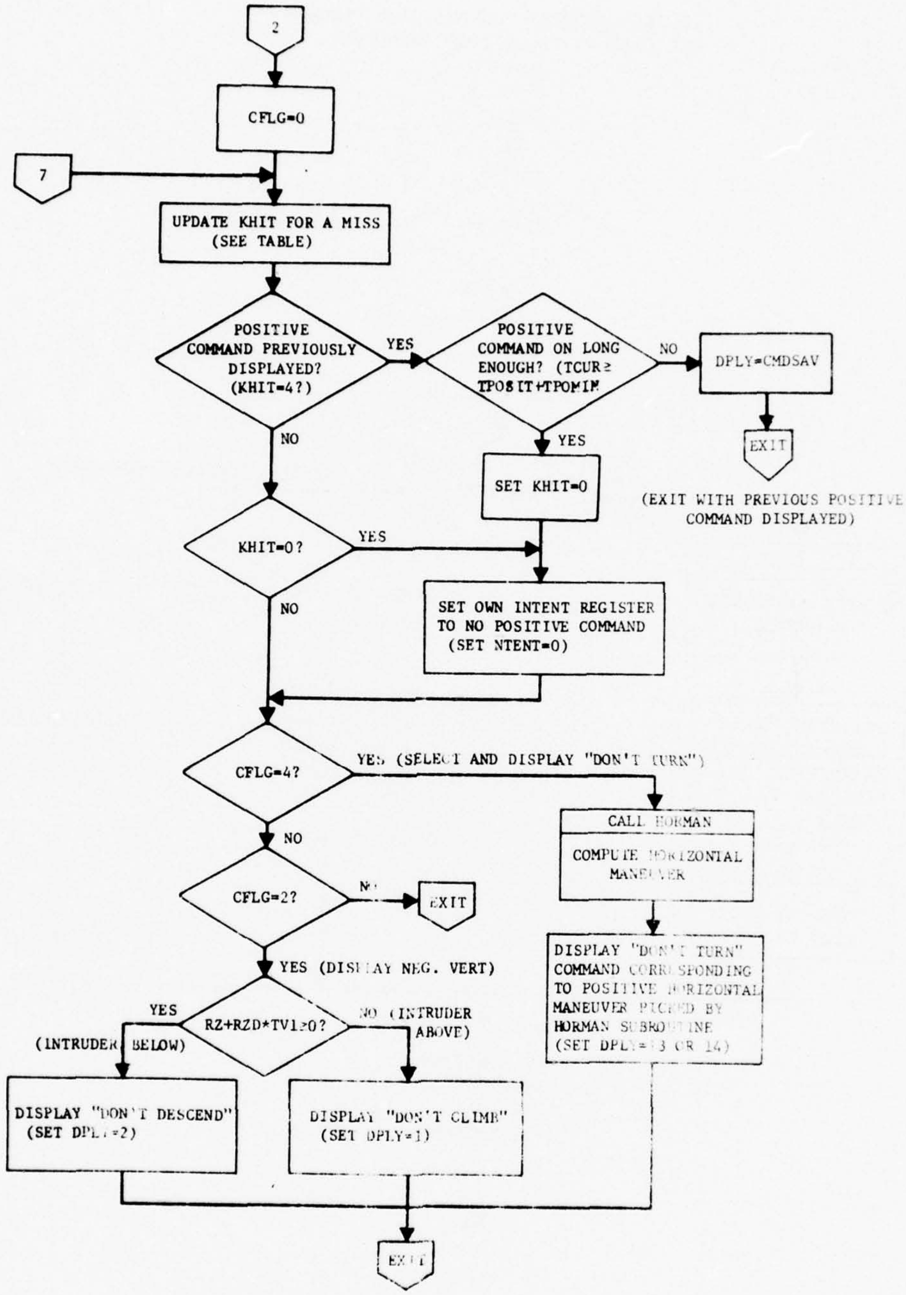


FIGURE 3-1 (Continued)
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS - PASSIVE MODE

NOTE: SUBSCRIPT 1 REFERS TO OWN AIRCRAFT, SUBSCRIPT 2 REFERS TO INTRUDER AIRCRAFT

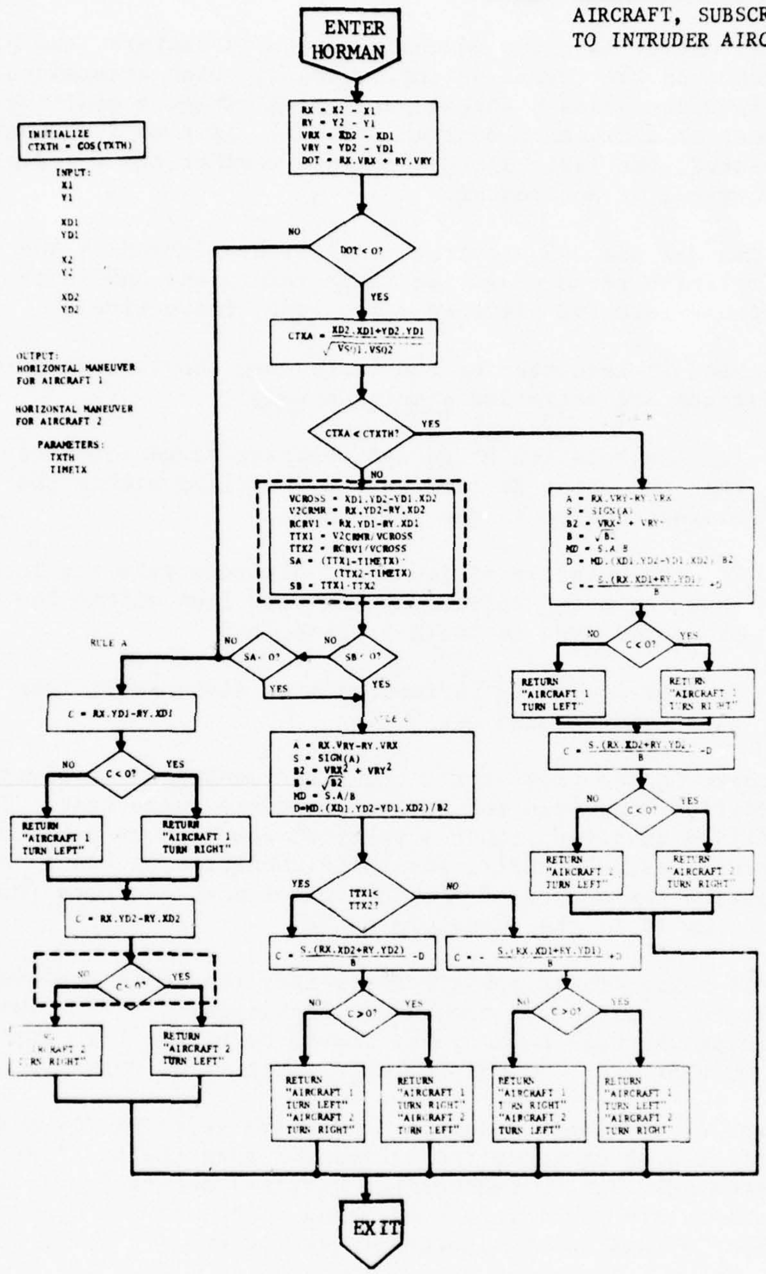


FIGURE 3-1 (CONCLUDED)
DETECTION AND RESOLUTION LOGIC FOR POSITIVE AND NEGATIVE COMMANDS--PASSIVE MODE

3.2 Threat Detection

After initializing the desensitization parameters, the program proceeds to the threat detection logic, which determines whether the intruder poses a threat warranting either a positive command request or a negative command request. If such a request is warranted, the logic also determines whether the maneuver should be vertical or horizontal.

Figures 3-2 and 3-3 show the protection afforded by the logic in the relative range--relative range rate plane and in the relative altitude--relative altitude rate plane, respectively.

A command is requested by the logic when the following three conditions are satisfied <u>simultaneously</u>:

1. the relative range and relative range rate are such that the point defined by the pair lies within the protected area in the R-$\dot{R}$ plane,

2. the relative altitude and altitude rate are such that the point defined by the pair lies within the protected area in the A-$\dot{A}$ plane, and

3. the projected horizontal miss distance is less than a certain threshold.

As shown in the flow chart, the first condition involves either a modified range-tau test or an immediate range test. The second condition involves either a vertical-tau test or an immediate altitude test. Finally, the third condition is tested by computing the square of the horizontal miss distance (MD2) and comparing it to the threshold MDCMD.

If all three conditions are satisfied, the logic proceeds to determine the type of command to be requested. This involves deciding whether the maneuver should be vertical or horizontal and whether the command should be positive or negative.

As shown, the logic requests a negative vertical command if the vertical miss distance VMD is greater than the limit ALIM2. The program sets VMD to the current vertical separation A if the aircrafts are vertically separating (i.e., if $\dot{A} > 0$) provided INDEX $\neq$ 3* and to the linear projection $A + \dot{A} *$ TVPCMD if the

---

* If INDEX=3, indicating extreme desensitization, and if the aircrafts are vertically separating, the positive command logic is desensitized further by setting the projected miss distance VMD = $A + \dot{A}$ * P(TAUR) where the predictive function P is defined in Appendix D.

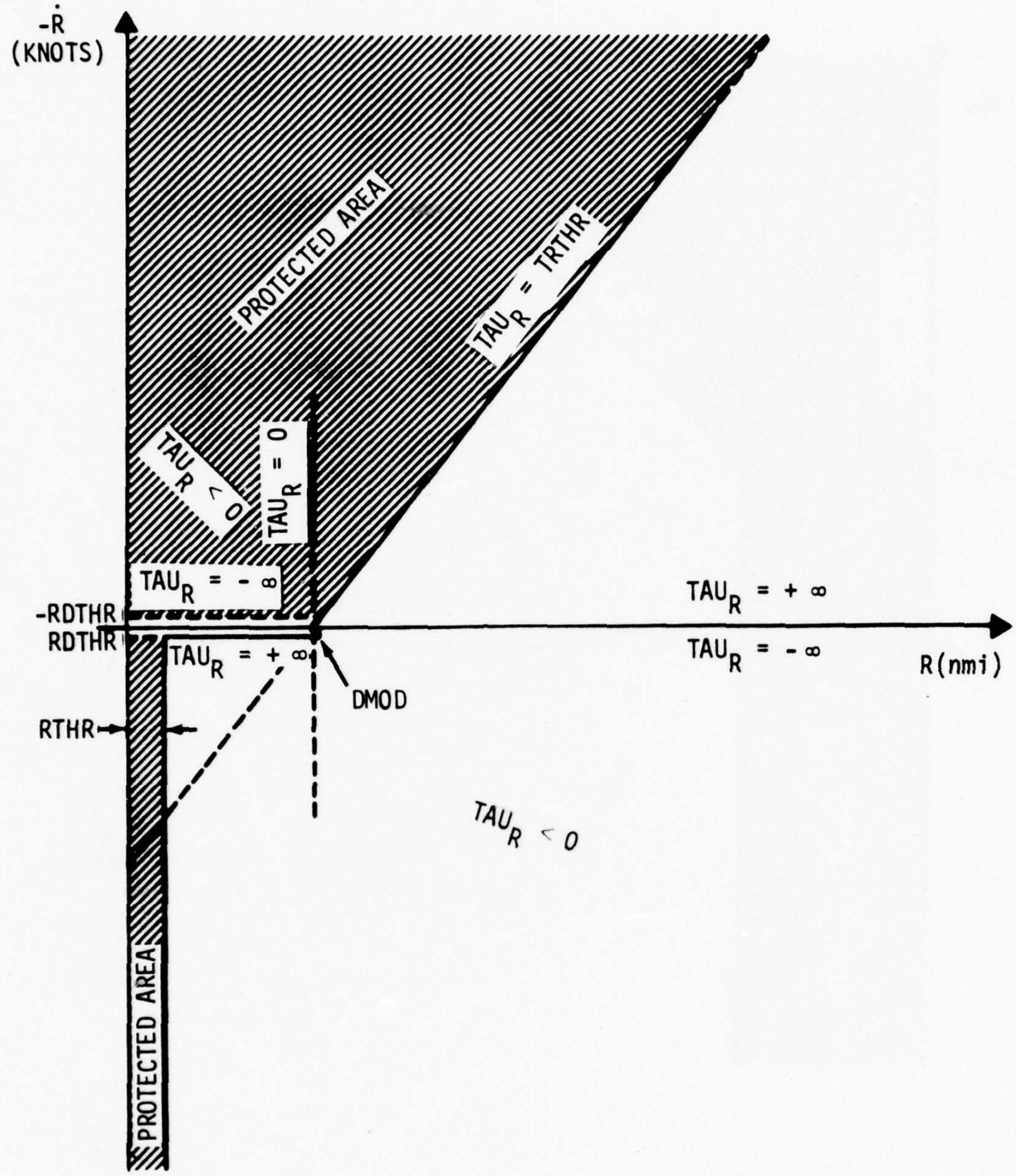


FIGURE 3-2
PROTECTION AFFORDED BY THE DETECTION LOGIC IN THE RELATIVE RANGE – RELATIVE RANGE RATE (R-Ṙ) PLANE

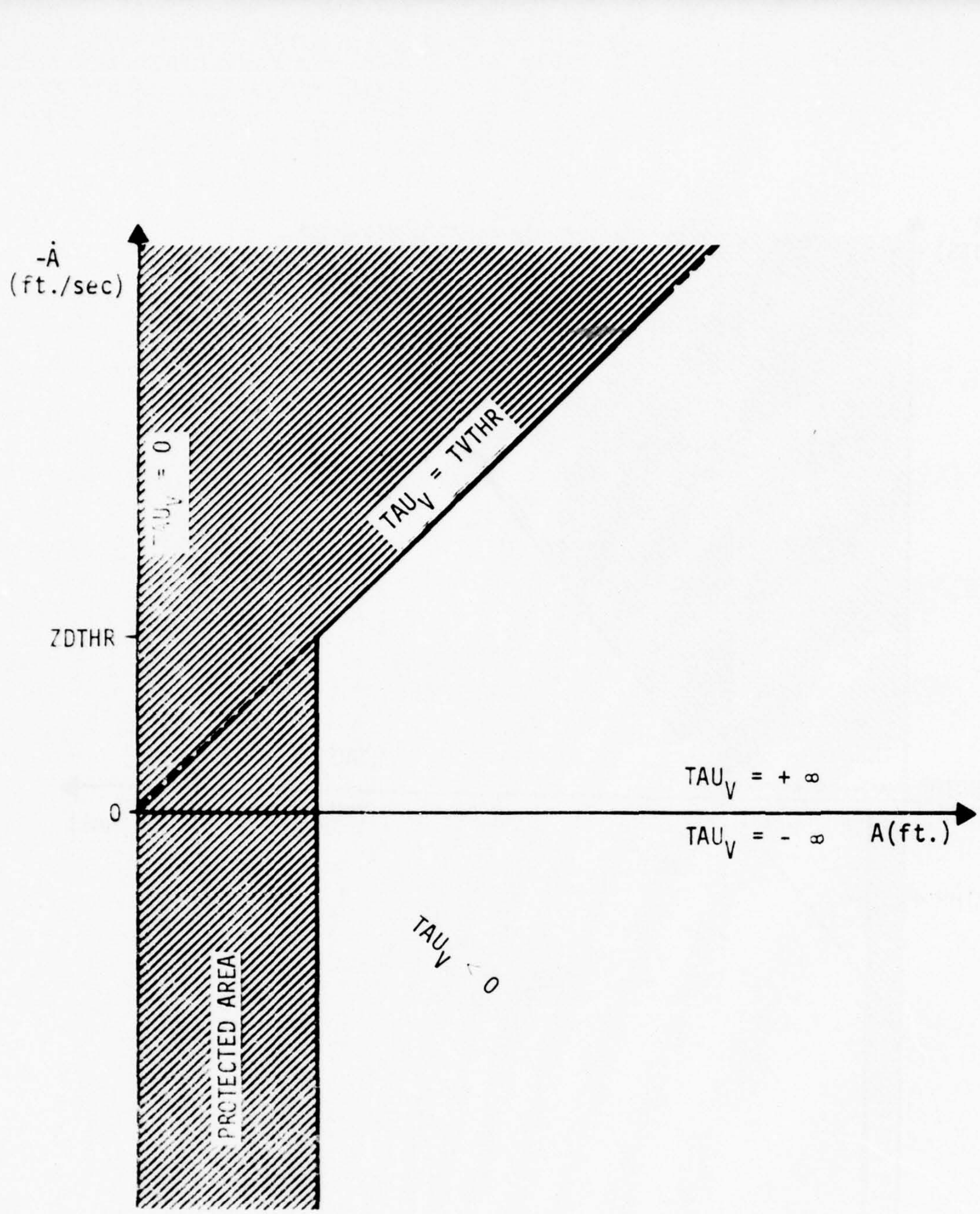


FIGURE 3-3
PROTECTION AFFORDED BY THE DETECTION LOGIC IN THE RELATIVE ALTITUDE – RELATIVE ALTITUDE RATE (A-Ȧ) PLANE

aircrafts are vertically closing. Note that, if $\dot{A} < 0$, VMD will be negative if the product of the rate $\dot{A}$ and the time TVPCMD is larger in magnitude than the current separation A (indicating that a vertical crossing occurs in the TVPCMD interval). If VMD $\geq$ ALIM2, the program sets CFLG=2, thereby requesting a negative vertical maneuver, and then exits the detection logic to process the request.

However, if the vertical miss distance is smaller than the limit, further tests are made to determine the type of request. At this juncture, the logic will select either a positive or a negative horizontal maneuver provided that the desired selection is not overriden by the switch NOHOR, which controls whether horizontal commands are permitted, or NOHPOS, which controls whether positive horizontal commands are allowed. If the switch NOHOR=0, the logic compares the horizontal miss distance* to a threshold and requests a negative horizontal maneuver if MD2 $\geq$ MDPOS. However, if MD2 < MDPOS, the program requests a positive horizontal maneuver provided that the override switch NOHPOS permits (i.e., if NOHPOS=0). If either NOHOR or NOHPOS overrides the desired horizontal selection, the logic defaults to a request for a positive vertical maneuver.

The horizontal override switches NOHOR and NOHPOS were added to the logic to permit flexibility in testing at NAFEC. It is anticipated that they will be held fixed during a simulation.

3.3 Positive Command Requests

As just seen, one possible output of the detection logic is a request for a positive command. The logic for processing such a request is shown immediately following the detection logic. It determines whether and when a positive command is displayed.

If a positive command was displayed on the previous scan (i.e., if KHIT=4), the program exits, leaving the display unchanged. But if no positive command was displayed, the logic is more complicated. The logic for initially displaying a positive command requires that the command be requested on two consecutive scans or on two scans separated by a third. Furthermore, in the case of an equipped intruder, the logic considers whether the positive command requested is compatible with that displayed by the intruder or being considered by him for display.

* Actually, the squared miss distance MD2 is used in the test since MD2 was previously computed.

The variable KHIT is used to implement the two-out-of-three logic for initially displaying a positive command. KHIT is initially set to zero prior to the first request and updated after each "hit" (i.e., request for a positive command) and after each "miss" (i.e., no request) according to the rules shown in Table 3-1. The five integer values that can be assumed by KHIT are explained in Table 3-2. Note that KHIT=3 indicates that the two-out-of-three condition has been satisfied but note that positive commands are not displayed until KHIT=4. In the case of the unequipped intruder, these last two states of KHIT are really identical; and as shown in the flow chart for this case, KHIT is automatically incremented from 3 to 4. However, in the case of the equipped intruder, the two states are distinct, and KHIT is not incremented from 3 to 4 until the coordination logic has been completed.

As shown, the case of the equipped intruder is treated separately from that of the unequipped intruder since the former requires special coordination logic. The equipped intruder case is discussed first.

3.3.1 Equipped Intruder Case

For the purpose of communicating intent and coordinating positive commands, each equipped aircraft is provided with an "intent register", which is really just an integer variable whose thirteen integer codes describe the aircraft's maneuver "*intent*". Positive integer codes indicate that the coordination logic has selected a positive command for display. Negative codes indicate that the command is still provisional and awaits coordination with the intruder. The intent register codes are explained in Table 3-3. Note that each negative integer code is the "mirror image" of the corresponding positive code except for its provisional status, for example, +2 indicates a descend, whereas -2 indicates a provisional descend. Thus, upgrading the intent register from provisional to permanent status involves only a sign change.

As shown, the first request for a positive command results in the selection of a provisional command. The program finds that NTENT=0 and branches to select a provisional command, testing CFLG to determine whether the command should be vertical or horizontal. If CFLG=3, the intent register NTENT is set to either a provisional climb (NTENT = -1) or a provisional descend (NTENT = -2) according to whether the intruder is projected to be below or above. But if CFLG=5, indicating a horizontal maneuver, the subroutine HORMAN is called to select the desired turns for both own aircraft and the intruder's aircraft, and

TABLE 3-1

NEW VALUE OF KHIT AS A FUNCTION
OF OLD VALUE AND HIT OR MISS

	NEW KHIT	
OLD KHIT	HIT	MISS
0	2	0
1	3	0
2	3	1
3	3	1
4	4	4

TABLE 3-2

THE VARIABLE KHIT AND ITS STATES

VALUES OF KHIT	MEANING
0	NO PREVIOUS COMMANDS, NO PREVIOUS HITS
1	A HIT TWO SCANS AGO AND A MISS THE PREVIOUS SCAN
2	A HIT ON THE PREVIOUS SCAN
3	2-OUT-OF-3 RULE HAS BEEN SATISFIED BUT POSITIVE COMMAND HAS NOT YET BEEN DISPLAYED
4	POSITIVE COMMAND IS DISPLAYED

TABLE 3-3

INTENT REGISTER CODES

CODE	MEANING
-6	PROVISIONAL OWN TURN RIGHT, INTRUDER TURN LEFT
-5	PROVISIONAL OWN TURN RIGHT, INTRUDER TURN RIGHT
-4	PROVISIONAL OWN TURN LEFT, INTRUDER TURN LEFT
-3	PROVISIONAL OWN TURN LEFT, INTRUDER TURN RIGHT
-2	PROVISIONAL DESCEND
-1	PROVISIONAL CLIMB
0	NO COMMAND SELECTED
1	CLIMB
2	DESCEND
3	OWN TURN LEFT, INTRUDER TURN RIGHT
4	OWN TURN LEFT, INTRUDER TURN LEFT
5	OWN TURN RIGHT, INTRUDER TURN RIGHT
6	OWN TURN RIGHT, INTRUDER TURN LEFT

NOTE: Negative integers indicate provisional commands, and a provisional command can be upgraded from provisional to permanent status simply by setting the minus sign to a plus.

NTENT is set to one of the four provisional horizontal codes. Before exiting, the program stores the provisional selection time, setting TPROV to the current time TCUR.

If on a subsequent scan the intent register is negative, indicating that a provisional command has already been selected, the logic tests KHIT to determine whether the two-out-of-three condition has been satisfied. The program proceeds to the coordination logic if KHIT=3 and exits otherwise. The coordination logic determines whether the provisional command in own intent register is displayed or whether some other positive command more compatible with the intruder's intent is displayed. This is determined by first testing whether a reply has been received from the intruder since the provisional selection time TPROV. If not, the program exits unless the reply wait time has been exceeded, in which case own intent is changed from provisional to permanent status and displayed. But, as shown, if a reply has been received since TPROV, the logic proceeds to test whether own intent code NTENT and the intruder's intent code MTENT are compatible.* If they are compatible, own intent NTENT is changed from provisional to permanent status (i.e., NTENT is set to -NTENT), and own intent determines the positive command displayed.

However, if own provisional intent is incompatible with the intruder's intent, further tests are needed to determine whose intent should take precedence. As shown, there are two additional cases in which own intent takes precedence. The first case occurs when the intruder's intent is also provisional (i.e., MTENT < 0) and own ID precedes the intruder's ID. The second case occurs when, simultaneously, the intruder's intent is permanent (i.e., MTENT > 0), own provisional intent is vertical (i.e., NTENT = -1 or -2), and the vertical miss distance VMD $\geq$ ACCEPT. In both these cases, own intent is changed from provisional to permanent status by reversing the sign of NTENT.

In all other cases where the intent registers are incompatible, the intruder's intent MTENT takes precedence. This includes all cases in which the intent is horizontal as well as the case in which the intent is vertical and the vertical miss distance VMD is small. In all of these cases, as well as the one mentioned earlier in which the intruder's ID takes precedence, the intruder's intent MTENT is used to redetermine own intent NTENT. This is accomplished using the absolute value |MTENT| as a

* See Appendix A for a discussion of compatibility.

pointer into the array ICOMP, which contains the compatible intent codes. Thus, NTENT is set to ICOMP (|MTENT|). The following codes are stored in the array ICOMP:

i	1	2	3	4	5	6
ICOMP(i)	2	1	6	4	5	3

As shown, the various branches of the coordination logic merge once the intent register is ugraded from provisional to permanent status. The program then uses the permanent intent to select the proper display code DPLY: NTENT is used as a pointer into the array NDPLY to set DPLY = NDPLY(NTENT).* Next, the program sets the positive command start time TPOSIT to the current time TCUR. Finally, just before exiting, the program sets KHIT=4.

3.3.2 Unequipped Intruder Case

As shown, processing a request for a positive command is much simpler in the case of the unequipped intruder since no coordination is necessary. If KHIT=2, indicating that the request is the first, the program exits. But if KHIT=3, indicating that this is the second request for a positive command, KHIT is automatically set to 4, and the program proceeds to determine the positive command to be displayed. If CFLG=3, a vertical maneuver is indicated, and the program sets the display code DPLY=5 or 6 to indicate a climb or a descend according to whether the intruder is projected to be below or above own aircraft. Also, the intent NTENT is set to 1 or 2 to indicate a climb or a descend, as the case may be. However, if CFLG≠3, a horizontal maneuver is indicated, and the program calls the subroutine HORMAN to determine the direction of the turns. The intent register is set to the proper positive code, and the display indicator DPLY is set to the turn selected. Regardless of whether a vertical or a horizontal maneuver is selected, the program sets TPOSIT to the current time TCUR and then exits.

3.4 Processing a "Miss"

A "miss" is said to have occurred if the detection logic does not request a positive command. As shown, the variable KHIT, which tracks the hit-miss and display status of positive commands, is updated to reflect the miss and then tested (KHIT=4?) to determine whether a positive command was displayed on the previous scan.

---

* The display codes stored in the array NDPLY are shown on the flow chart.

If KHIT=4, the logic tests whether the command has been displayed for the minimum length of time required, namely, TPOMIN seconds. If not, the program exits with the positive command still displayed. But if the current time TCUR $\geq$ TPOSIT + TPOMIN, KHIT and the intent register are both reset to zero, thereby wiping-out the command and re-initializing the two-out-of-three logic described previously.

If KHIT < 4, the logic tests whether KHIT=0. If so, the intent register is reset to zero.

The logic just described for processing a miss is entered when a negative command is requested by the detection logic as well as when no command is requested. The subsequent processing of a request for a negative command is discussed in the next subsection.

3.5 Negative Command Requests

After processing the miss, the program proceeds to test CFLG to determine whether the detection logic requested a negative command (provided the program did not exit due to the condition described in the previous subsection). If CFLG=4, a negative horizontal maneuver (i.e., "don't turn") was requested. The direction of the "don't turn" is inferred from the positive maneuver selected by the subroutine HORMAN. For example, if HORMAN selects "turn right", the display indicator DPLY is set to 14 to indicate "don't turn left". If CFLG=2, a negative vertical maneuver was requested, in which case DPLY is set to 1 or 2 to display either a "don't climb" or a "don't descend" according to whether the intruder is projected to be above or below own aircraft. The program exits after setting the display indicator to the desired negative command.

3.6 Computation of Horizontal Maneuvers

The computation of the best pair of positive horizontal maneuvers for the intruder and own aircraft is shown in the flow chart for the subroutine HORMAN. The logic was borrowed from IPC, and except for the addition of the entry point "ENTER HORMAN", the flow chart was taken from Reference 2. The reader interested in a more detailed discussion of the horizontal resolution algorithm is referred to that document.

3.7 Detection and Resolution Logic for the Active Mode

The flow chart for the active mode logic (DRACT) is presented in Figure 3-4. As the flow chart shows, the detection and resolution logic for the active mode is essentially a simplification of the

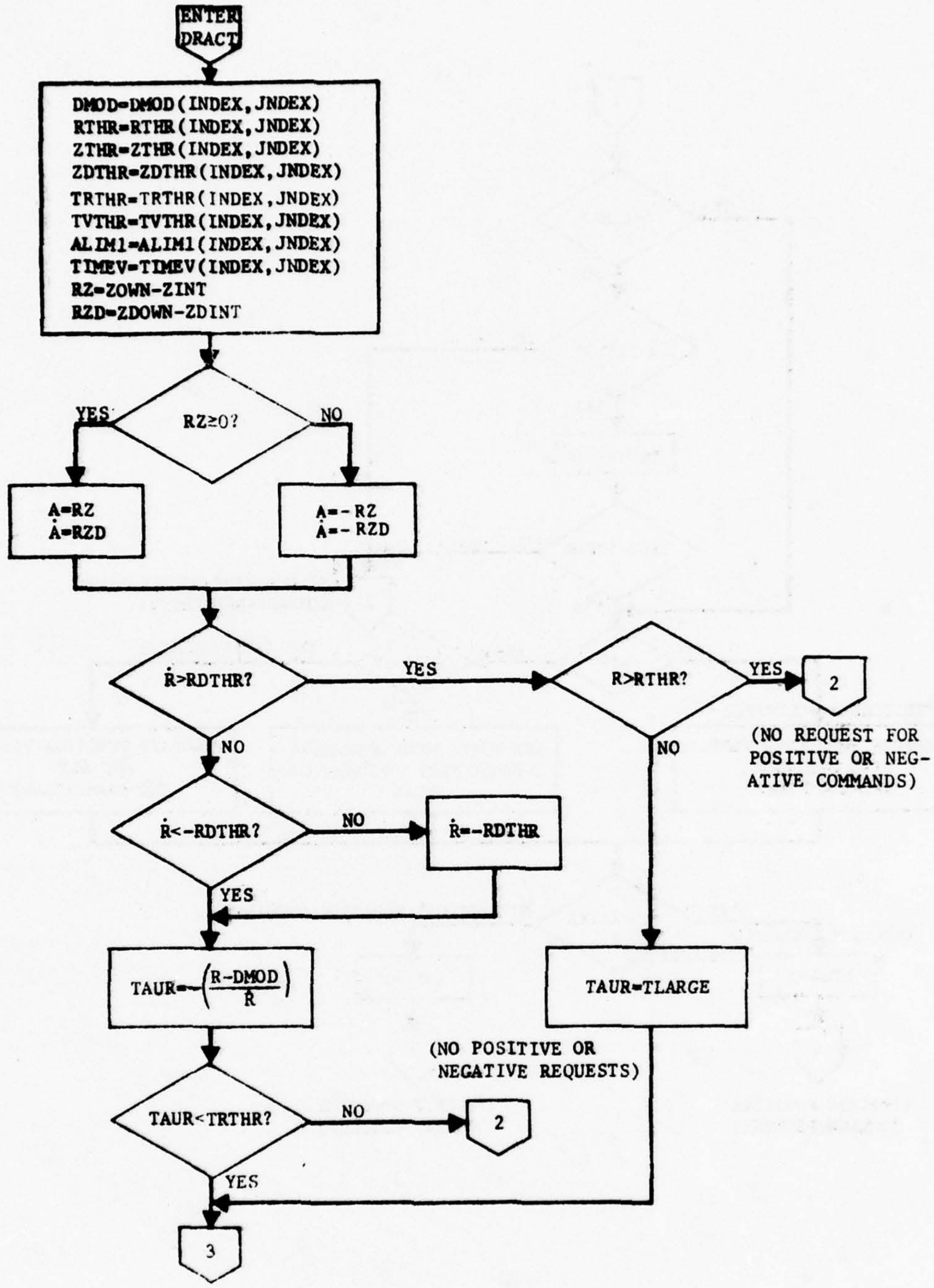


FIGURE 3-4
DETECTION AND RESOLUTION FOR POSITIVE AND NEGATIVE COMMANDS – ACTIVE MODE

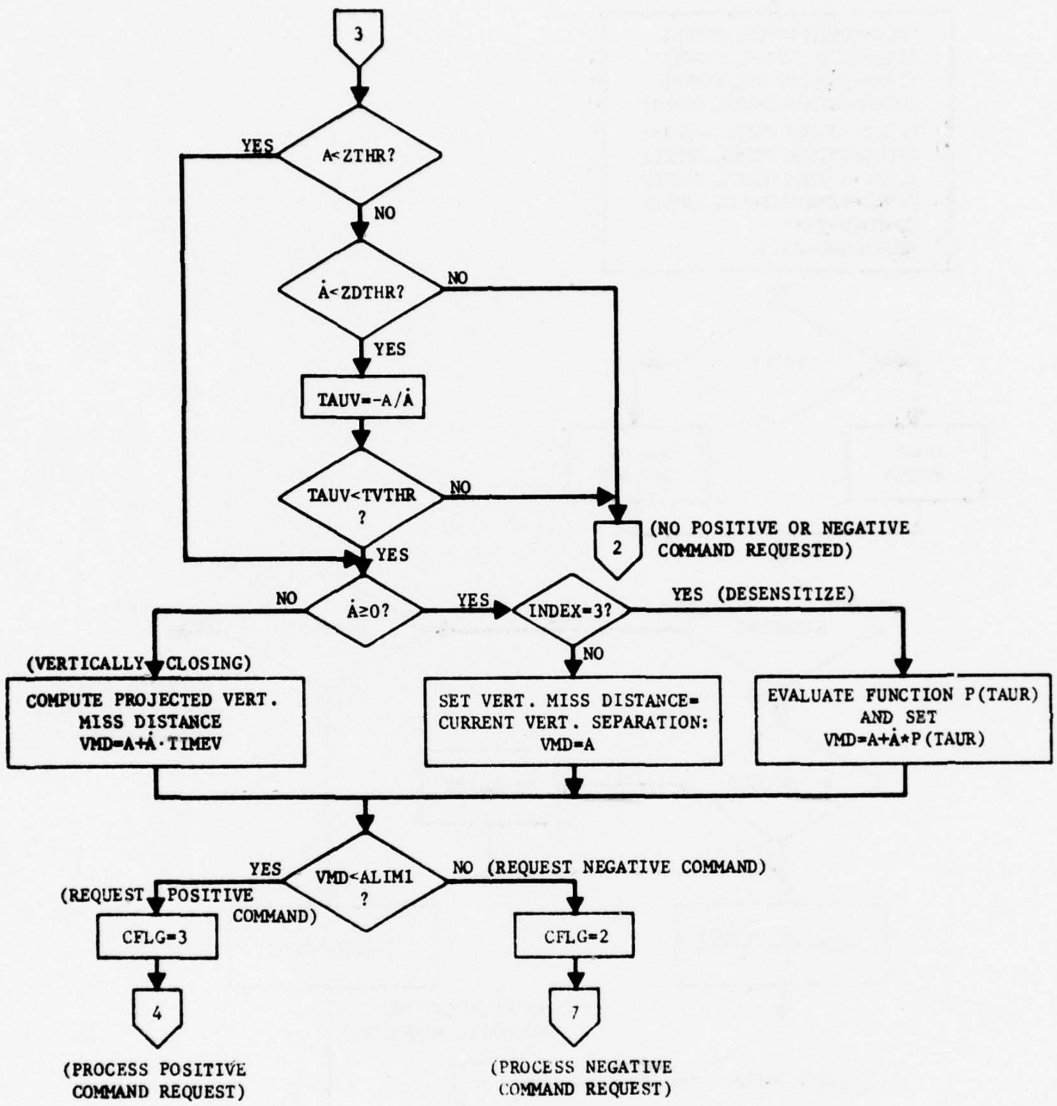


FIGURE 3-4 (Continued)
DETECTION AND RESOLUTION FOR POSITIVE AND NEGATIVE COMMANDS – ACTIVE MODE

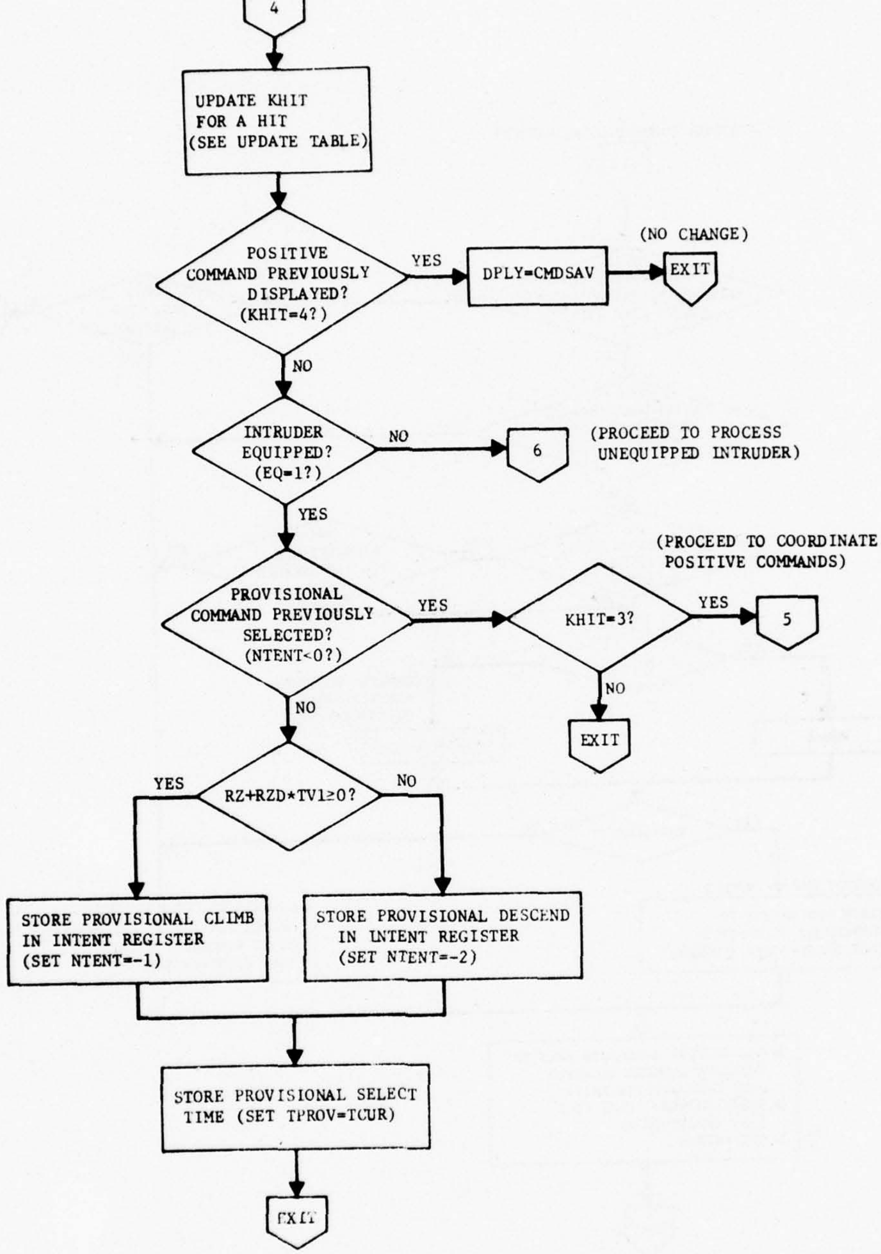


FIGURE 3-4 (Continued)
DETECTION AND RESOLUTION FOR POSITIVE AND NEGATIVE COMMANDS – ACTIVE MODE

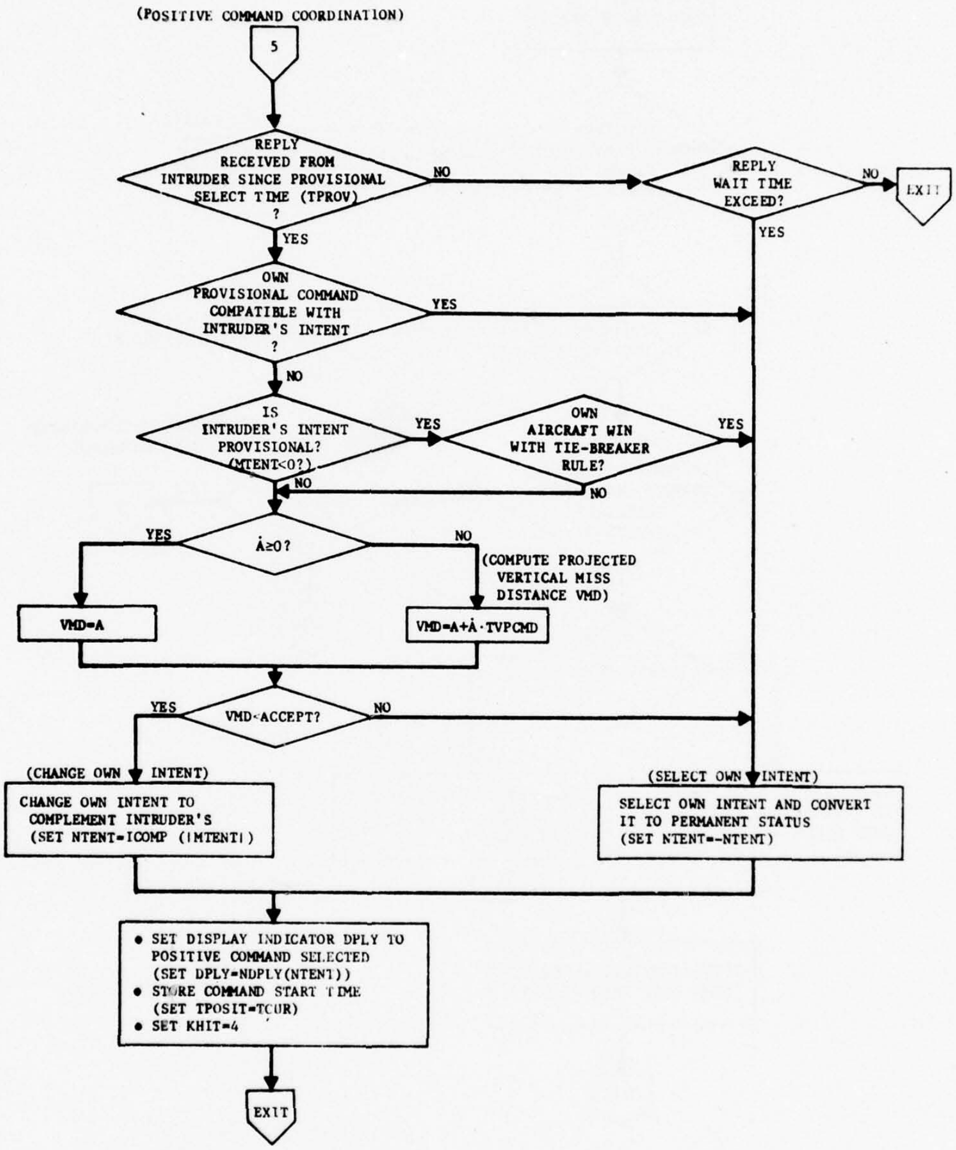


FIGURE 3-4 (Continued)
DETECTION AND RESOLUTION FOR POSITIVE AND NEGATIVE COMMANDS – ACTIVE MODE

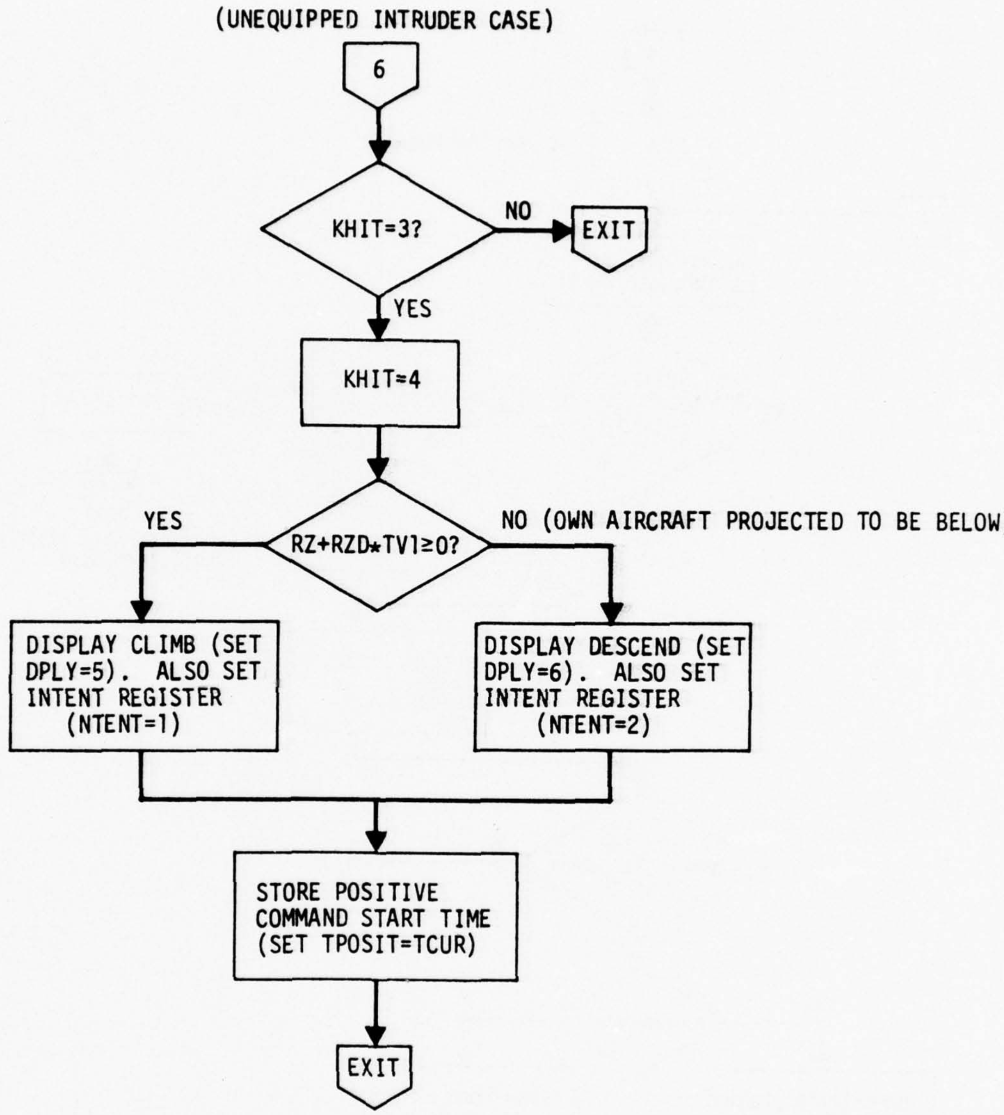


FIGURE 3-4 (Continued)
DETECTION AND RESOLUTION FOR POSITIVE AND NEGATIVE COMMANDS – ACTIVE MODE

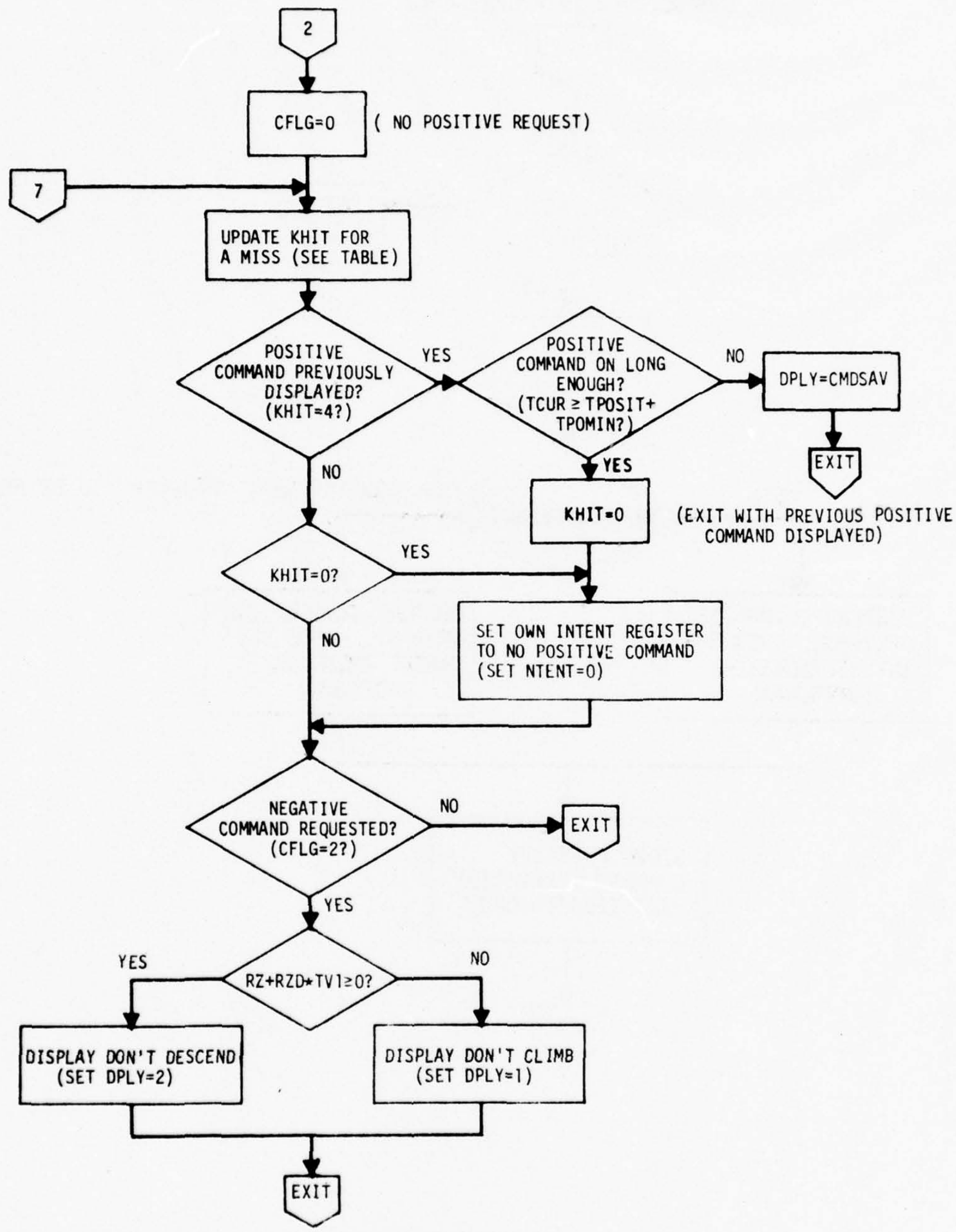


FIGURE 3-4 (Concluded)
DETECTION AND RESOLUTION FOR POSITIVE AND NEGATIVE COMMANDS – ACTIVE MODE

logic for the passive system described above. The active logic is simpler primarily because only vertical collision avoidance maneuvers are considered. Thus, the conflict resolution logic is considerably simplified.

This section does not discuss the logic in detail since the discussion would only be an abbreviated repetition of the previous section. The flow chart of the active logic should require no additional explanation for the reader familiar with the previous section's discussion of the passive logic. As shown, the desensitization logic is identical except for the deletion of two parameters (viz., MDCMD and MDPOS), the substitution of ALIM1 for ALIM2, and the addition of the parameter TIMEV. The detection logic is the same except for the absence of two horizontal miss distance tests and the logic for requesting horizontal commands.

4. LIMIT VERTICAL RATE COMMAND LOGIC

Figure 4-1 presents the flow chart of the logic which determines whether a vertical limit command should be displayed. This subroutine is only executed if limit commands have been selected with the LIMCOM option switch and if no positive or negative commands have been requested for this intruder. Note that DPLY is always zero when entering this subroutine. If no limit commands are required, then DPLY is zero on exiting the subroutine. Otherwise, DPLY represents the limit command to be displayed as coded in Appendix C.

If the vertical separation A is very large (i.e., if $A > LALT$), limit commands are not needed, and the program exits. But if $A \leq LALT$, the program computes the modified-tau variable TAU2 and compares it to the parameter TAU2L as a further test of whether a limit command is required. If not, the program exits with the display indicator DPLY = 0 to indicate "no command".

However, if the test shows the need for a limit command, the program proceeds to compute the proper limit and its direction (i.e., up or down). If the vertical separation A does not exceed BAND1, the vertical separation rate is limited to a maximum of 500 feet/minute. If A is between the limits BAND1 and BAND2, the rate is limited to a maximum of 1000 feet/minute. Otherwise, A is between BAND2 and LALT, in which case the rate is limited to a maximum of 2000 feet/minute. The limit command is either a "limit climb" or a "limit descent" to the rate determined by the vertical separation A, the direction being determined by whether the intruder is above or below own aircraft. Before exiting, the program sets DPLY to the display code corresponding to the limit command just selected.

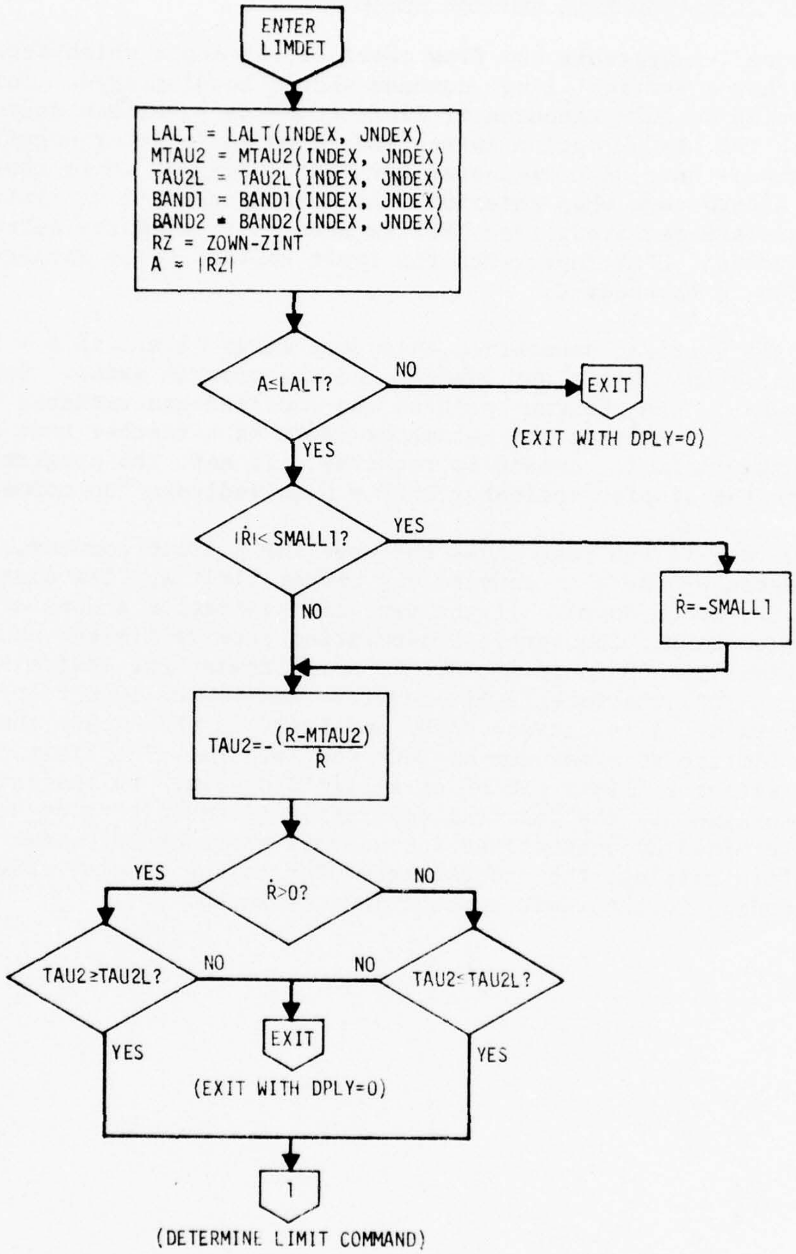


FIGURE 4-1
LOGIC FOR DETERMINING LIMIT COMMANDS (LIMDET)

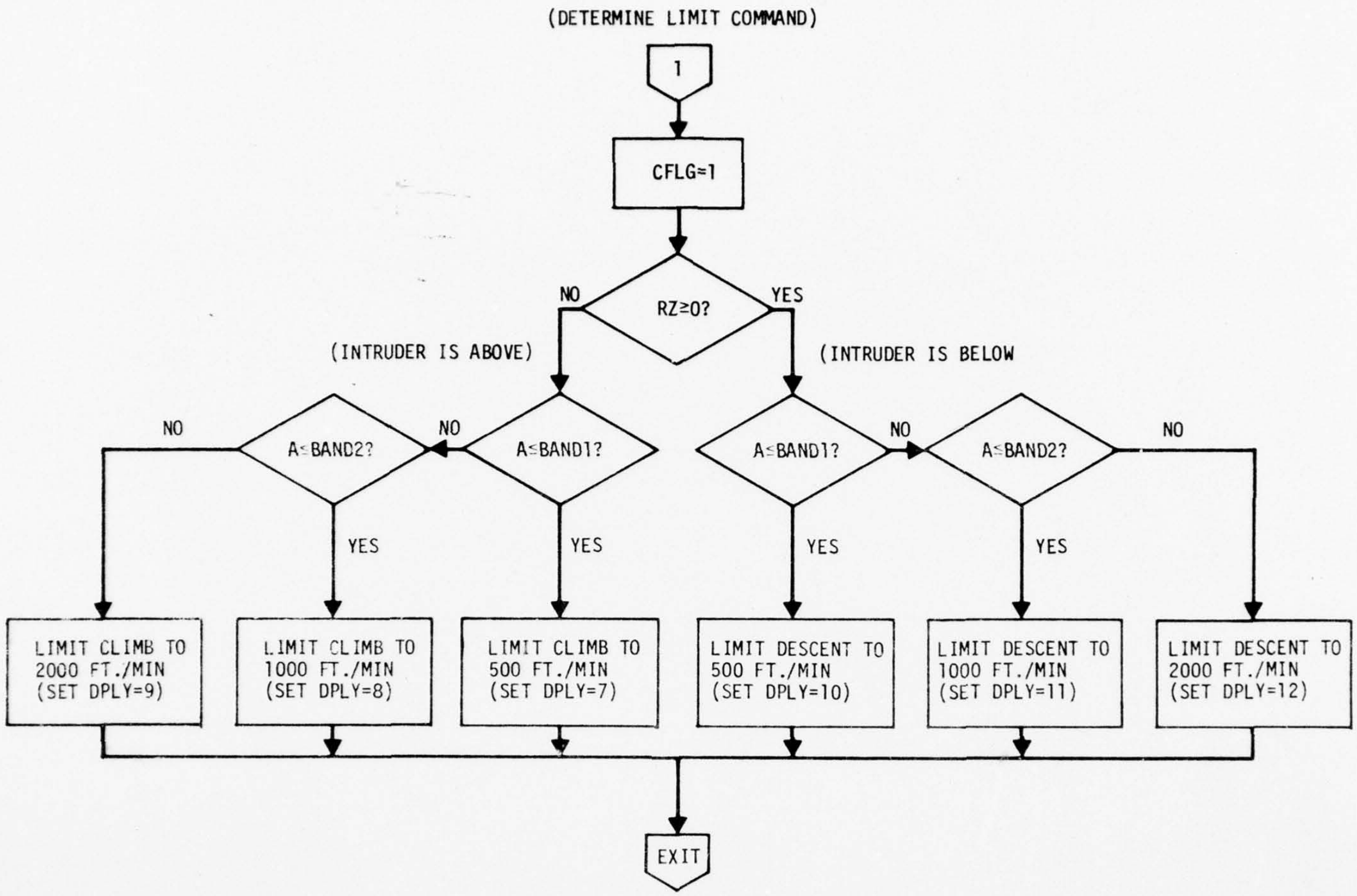


FIGURE 4-1 (Concluded)
LOGIC FOR DETERMINING LIMIT COMMANDS (LIMDET)

5. INTRUDER POSITION DATA LOGIC

This section presents the IPD logic that determines whether or not position data will be displayed for a particular intruder. The flow chart for this subroutine is presented in Figure 5-1. The subroutine is executed only if either the CLIPD or PVIPD option is selected. The IPD logic is not dependent upon any calculations performed in the command detection logic. Therefore, an IPD only display can be implemented by turning the PNCOM and LIMCOM switches off. However, the IPD logic should not be used with the active mode logic, since bearing to the intruder is needed to display intruder position data.

The IPD calculations involve a number of parameters determined by the desensitization level, namely, DMODP, RTHPO, RTHPF, TIPDO, TIPDF, RZIPDO, RZIPDF, and MDIPDF. Nominal values of these parameters are listed in Appendix B. As with other parameters used in the command detection logic, these parameters are set as a function of the desensitization level, represented by INDEX, and the equipage of the intruder, represented by INDEX. INDEX is set in the own data tracking subroutine, and JNDEX is set in the intruder tracking subroutine. The IPD subroutine outputs the 2 flags IPDFLG and FLSHFL. If it is set, the IPDFLG flag will cause position data for this intruder to be displayed. If FLSHFL is set, the intruder position symbol will flash; otherwise it will remain steady. It is intended that the threshold for display of the ordinary IPD data define larger protection volumes than those for the display of flashing IPD data. A horizontal miss distance test is a part of the flashing IPD test but not of the ordinary IPD test.

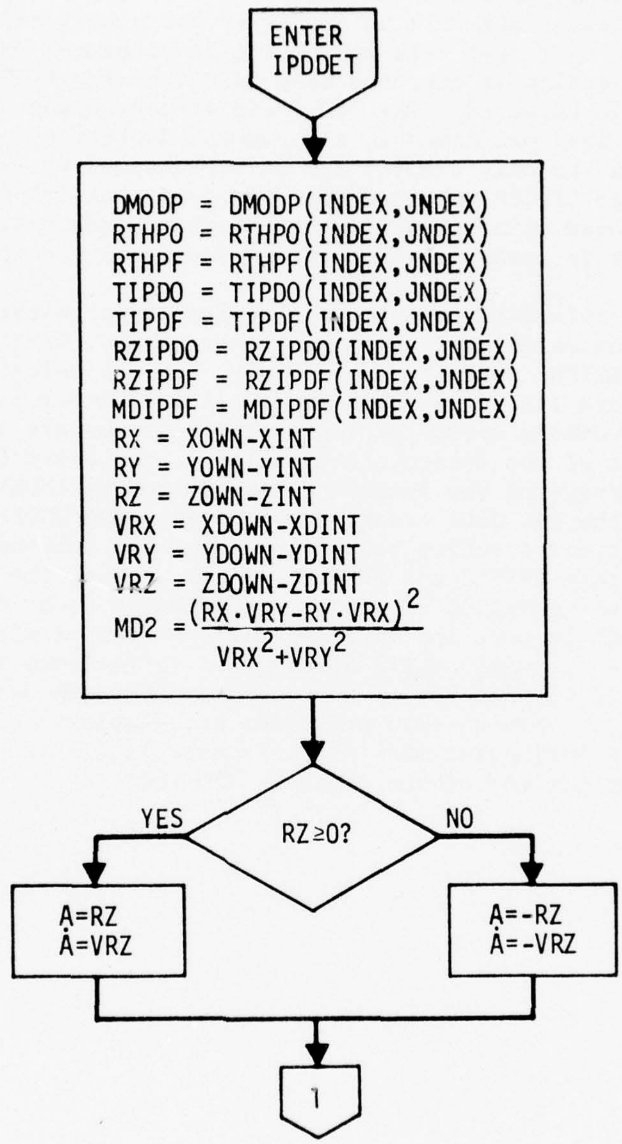


FIGURE 5-1
INTRUDER POSITION DATA DETECTION LOGIC (IPDDET)

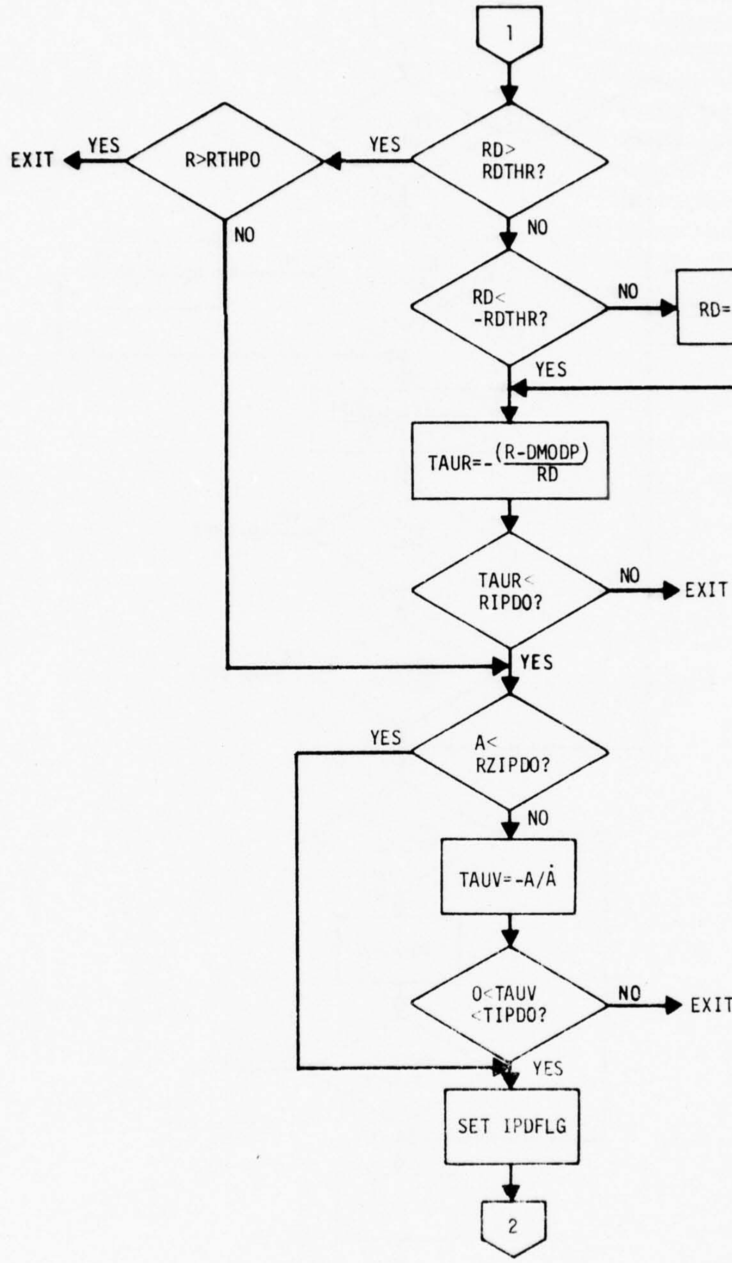


FIGURE 5-1 (Continued)
INTRUDER POSITION DATA DETECTION LOGIC (IPDDET)

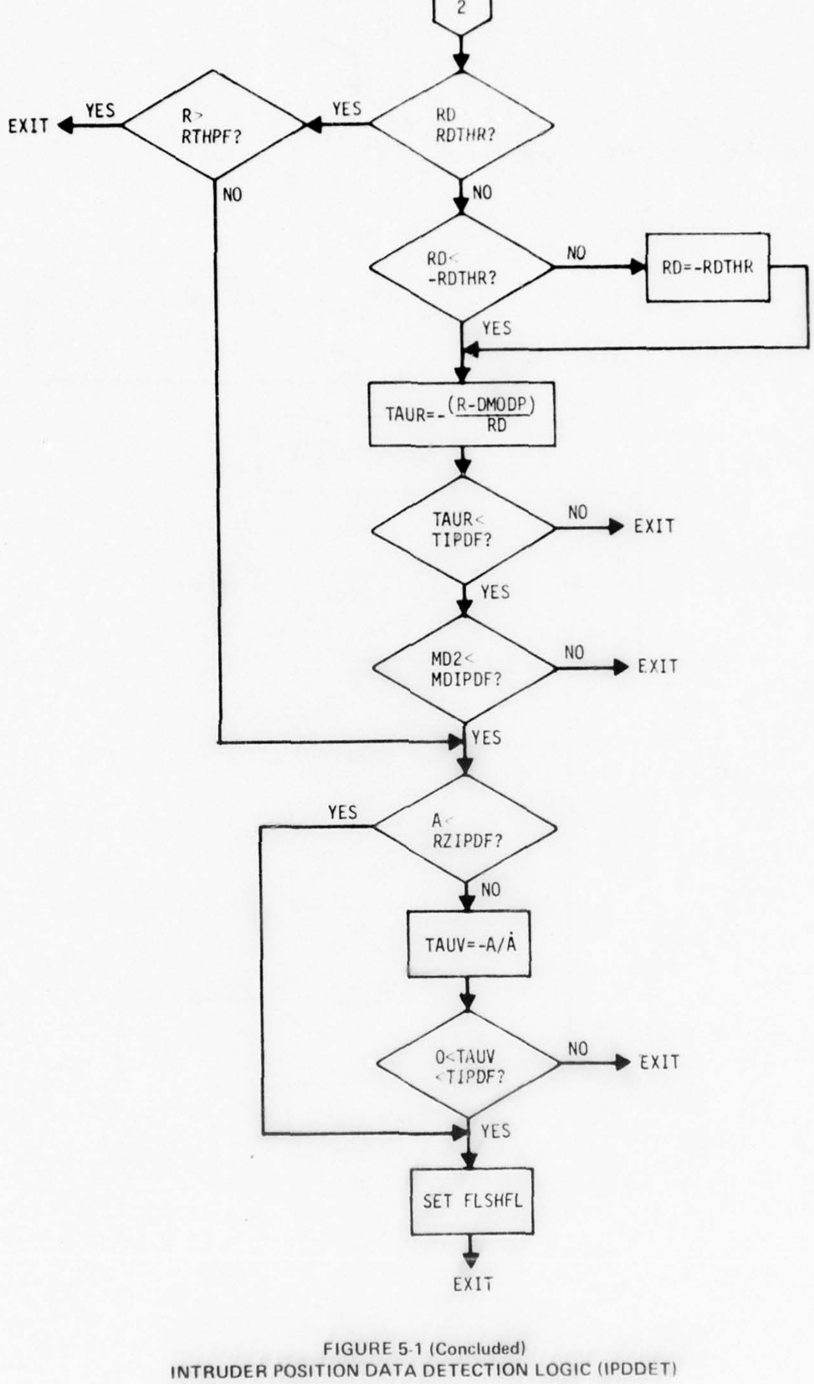


FIGURE 5-1 (Concluded)
INTRUDER POSITION DATA DETECTION LOGIC (IPDDET)

6. DISPLAY LOGIC

The logic presented in this section generates the output which drives the BCAS display. The flow chart for this display logic is given in Figure 6-1. The logic flow passes through the display subroutine, DISPLY, every time an intruder is presented as a potential threat, regardless of the setting of any of the option switches. The subroutine DISPLY is used for both active and passive mode operation and to drive any of the displays that might be tested with the BCAS logic.

Three types of displays are expected to be tested (at different times) using this logic, namely, the ACAS display, the IPD display, and the PVD display. The Airborne Collision Avoidance System (ACAS) display, described in reference 1, is capable of displaying CLIMB, DESCEND, DON'T CLIMB, DON'T DESCEND commands and limit vertical rate commands. It cannot display intruder position data. The Intermittent Positive Control (IPC) display, which is described in reference 2, can display positive or negative horizontal or vertical commands. In addition, it can display intruder position data expressed in own-heading oriented clock position and qualitative relative altitude information (intruder below, at same level or above). The third display that can be driven is an own-heading-oriented plan view display (PVD) in which one or more intruders can be presented at the correct relative bearing and range with additional character data presented in a data block attached to the intruder's position symbol.

A single output display vector is used to interface with all of the displays. This is shown in Table 6-1. The fields that may be read by any of the displays are indicated. Several intruders may be represented by IPD's at one time but only one intruder at a time can cause a positive, negative or limit command to be displayed. Eventually, the logic will be developed to include multiaircraft resolution and display capability. For the moment, once an intruder has claimed the command field with a positive, negative, or limit command, no positive, negative or limit command can be displayed by another intruder until the claiming intruder ceases to display commands.

The flag, COMACT, indicates whether or not the data in the command field is to be read and displayed by the display. When this flag is off, an IPD can be displayed on an intruder without wiping out a command being displayed for another intruder.

Non-null display vectors generated by the subroutine DISPLY are linked to a list of display vectors for own aircraft using

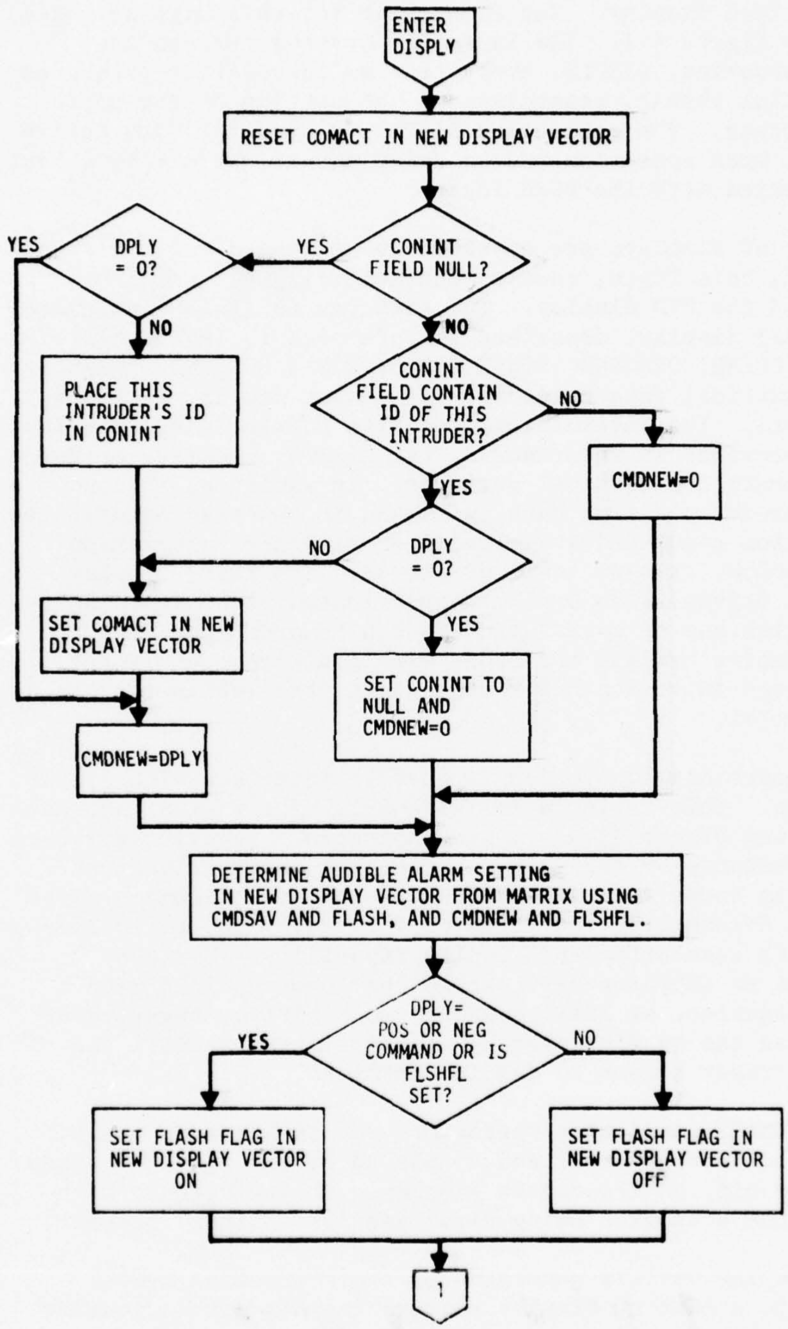


FIGURE 6-1
DISPLAY LOGIC (DISPLY)

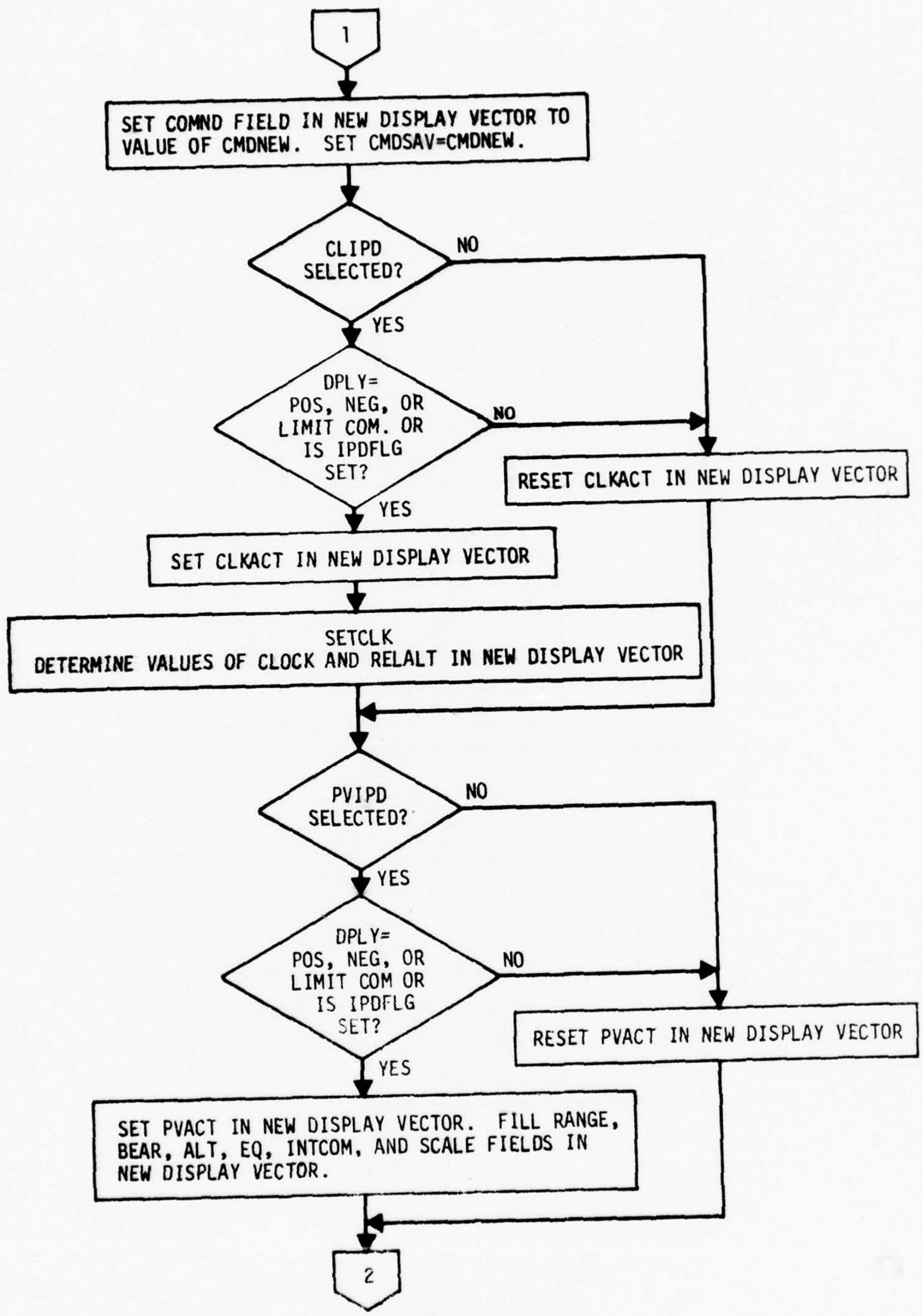


FIGURE 6-1 (Continued)
DISPLAY LOGIC (DISPLY)

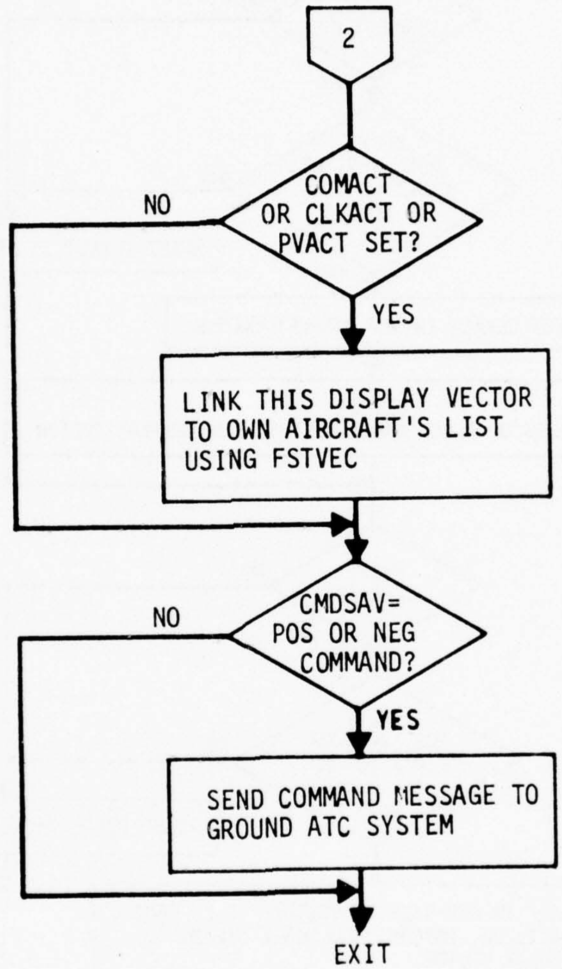


FIGURE 6-1 (Concluded)
DISPLAY LOGIC (DISPLY)

TABLE 6-1

OWN-AIRCRAFT DISPLAY VECTOR USED TO DRIVE
ALL TYPES OF DISPLAYS

Symbol	Meaning	Used by		
		ACAS	IPC	PVD
AUD	Audible alarm flag	X	X	X
FLASH	IPD flash flag		X	X
COMACT	Command field active flag	X	X	X
COMND*	Positive, negative, or limit command variable	X	X	X
CLKACT	Clock IPD active flag		X	
CLOCK	Clock position of IPD		X	
RELALT	Relative altitude of IPD		X	
PVACT	Plan view display IPD active flag			X
RANGE	Range to intruder			X
BEAR	Bearing to intruder (relative to own-heading)			X
ALT	Altitude of intruder			X
EQ	Equipage of intruder			X
INTCOM	Command of intruder			X
SCALE	Display scale used by PVD			X
NXTVEC	Pointer to next display vector	X	X	X

* The coding of this variable is the same as for DPLY and is given in Appendix C.

FSTVEC. At the end of the BCAS cycle, the display is wiped clean and then rewritten using all of the new display vectors simultaneously. Thus, the display is refreshed once per cycle. The list of display vectors is then zeroed and the accumulation of display vectors begins again on the next cycle.

When the flag CLKACT in the display vector is set, the variables CLOCK and RELALT contain meaningful data. The flag PVACT serves the same function for the variables RANGE, BEAR, ALT, EQ, INTCOM, and SCALE.

The first step in the flow chart of Figure 6-1 is to determine the command that will be displayed for this intruder on this cycle. This is represented by the local variable CMDNEW. Next, the audible alarm flag in the display vector is determined using a look-up table. One index into the table is obtained from the previous display state for the intruder as determined by CMDSAV and FLASH, both of which are stored in the intruder state vector. The other index is obtained from current display information, specifically from CMDNEW and FLSHFL, which are computed by the IPD logic. The table is presented as Table 6-2.

No audible alarm is given for limit commands. An audible alarm sounds if:

1. a transition from no command to a positive or negative command takes place,

2. a transition from a negative to a positive command takes place,

3. a transition from one positive command to another positive command or from one negative command to another negative command takes place,

4. a transition from no flashing IPD to a flashing IPD takes place.

Note that only data applicable to a single intruder goes into this decision. The existence of commands or IPD's for other intruders does not prevent sounding the audible alarm for one of the above events for the given intruder.

Next, the FLASH flag in the display vector is set. If IPD's are selected, the intruder's position symbol flashes if positive, negative, or limit commands were desired, or if the IPD tests indicated the need for a flashing IPD. Note that this test is based on the commands desired (DPLY), not on the commands to be displayed (CMDNEW).

TABLE 6-2

MATRIX GIVING SETTING OF AUDIBLE ALARM FLAG AS A FUNCTION OF OLD AND NEW DISPLAY CONDITIONS

New CMDNEW \ Old CMDSAV	NEW FLSHFL \ Old FLASH	Pos. Command	Neg. Command	No Pos. or Neg. Command	No Pos. or Neg. Command
		Any	Any	Off	On
Pos. Com.	Any	1	On	On	On
Neg. Com.	Any	Off	1	On	On
No Pos. or Neg. Com.	Off	Off	Off	Off	Off
No Pos. or Neg. Com.	On	Off	Off	On	Off

1. Audible alarm off if new command is same as old command, on if different.

When the clock IPD's are to be displayed, the clock position and relative altitude are determined from the flow chart in Figure 6-2. When plan view IPD's are being displayed, the data, except for SCALE, is drawn from the intruder's state vector. The value of SCALE is determined by SLEVEL in own aircraft state vector. Each value of SLEVEL will have a corresponding value of SCALE. SCALE is used to determine the display scale of the plan view IPD's.

As a last task, this subroutine generates a command message for transmittal to the ground ATC system. The content of this message is merely own aircraft's identification and own aircraft's displayed positive or negative command.

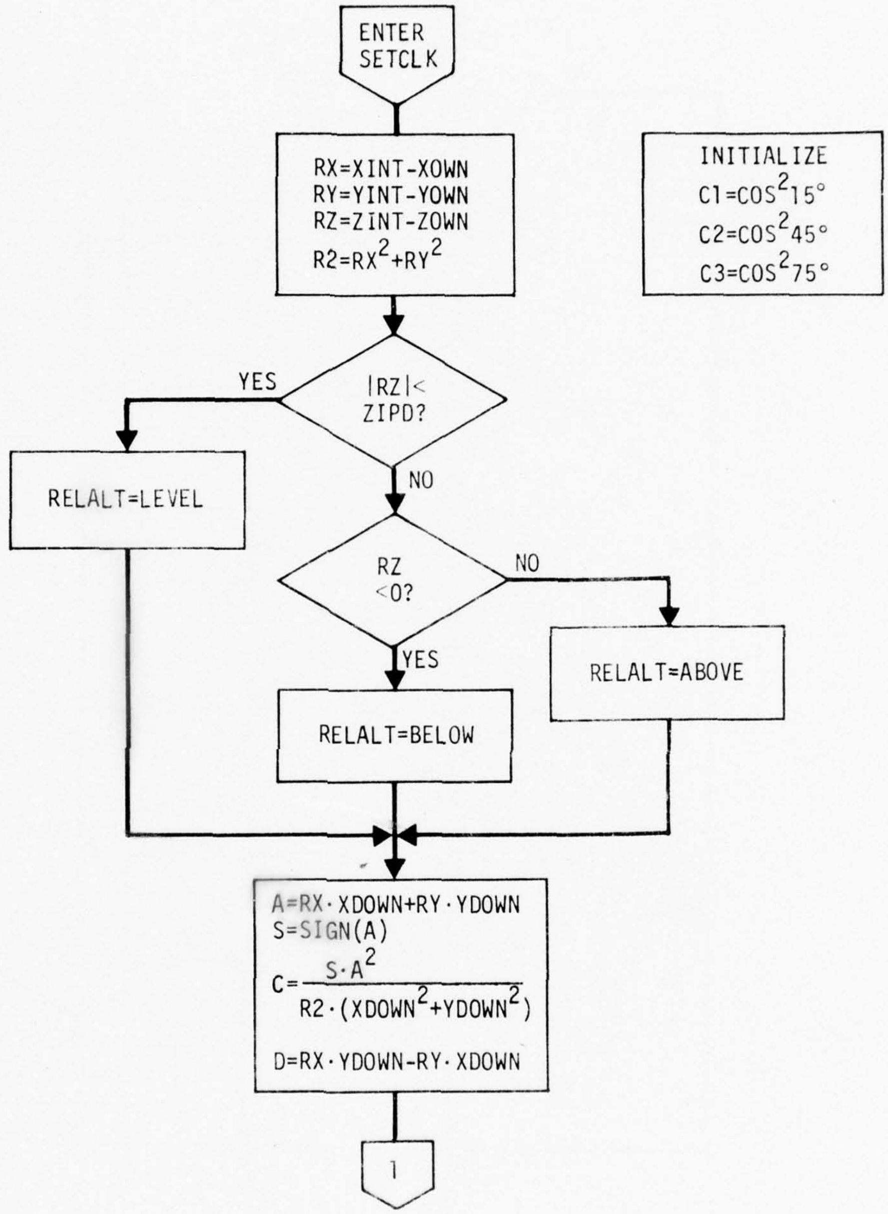


FIGURE 6-2
DETERMINING RELATIVE ALTITUDE AND CLOCK POSITION FOR AN IPD (SETCLK)

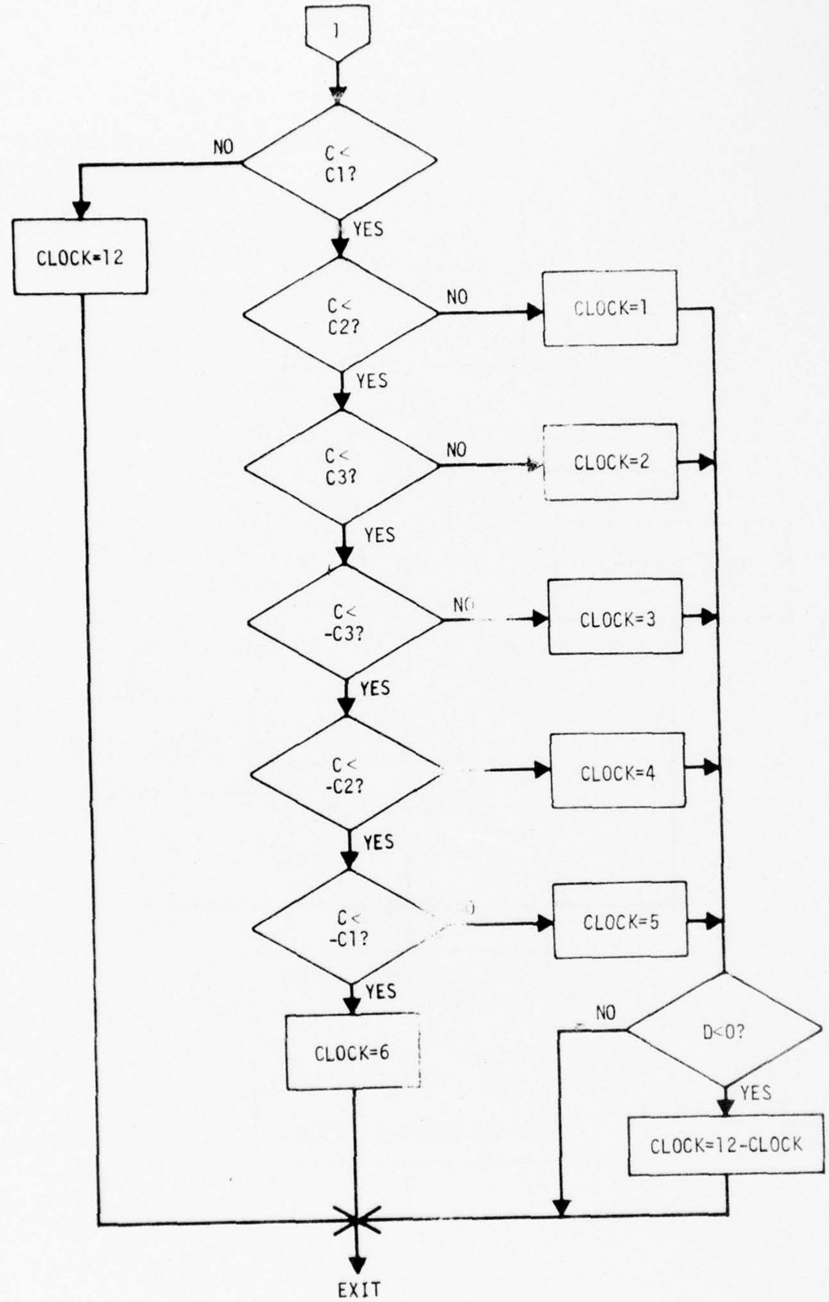


FIGURE 6-2 (Concluded)
DETERMINE RELATIVE ALTITUDE AND CLOCK POSITION FOR AN IPD (SETCLK)

7. TRACKING LOGIC

This section presents the tracking logic used by the BCAS collision avoidance logic. Separate flow charts are presented for the tracking logic that is to be used with the active mode logic and the passive mode logic.

Tracking is performed separately for own aircraft data and for data applying to intruders observed by own aircraft. Own aircraft data is tracked once per BCAS logic cycle whether or not own aircraft is in potential conflict with another aircraft. The intruder data for a given intruder is tracked when that intruder is presented to own aircraft as a potential threat. From the overall simulation point of view, all tracked data is duplicated. For instance, one aircraft will maintain its own tracked Z and will maintain a tracked Z on a specific intruder. At the same time, that intruder will maintain its own tracked Z and an intruder's tracked Z on the first aircraft. However, the tracked values may not always be exactly the same because missed reports are simulated separately for the two aircraft.

There are four tracking routines presented below. They are:

1. Tracking own data for the passive mode
2. Tracking own data for the active mode
3. Tracking intruder data for the passive mode
4. Tracking intruder data for the active mode

All tracking is accomplished with simple, fixed parameter tracking.

7.1 Tracking Own Data for the Passive Mode (TROPAS)

The flow chart for this subroutine is presented in Figure 7-1. The arguments in the subroutine call are given in Table 7-1.

In this flow chart it is assumed that no missed reports will be simulated for own report data. Furthermore, it is assumed that, when a new aircraft appears in the simulation, the external simulation will create an own-aircraft state vector and initialize it properly. Likewise, it is assumed that the external program will eliminate the own-aircraft state vector when an aircraft leaves the simulation scenario.

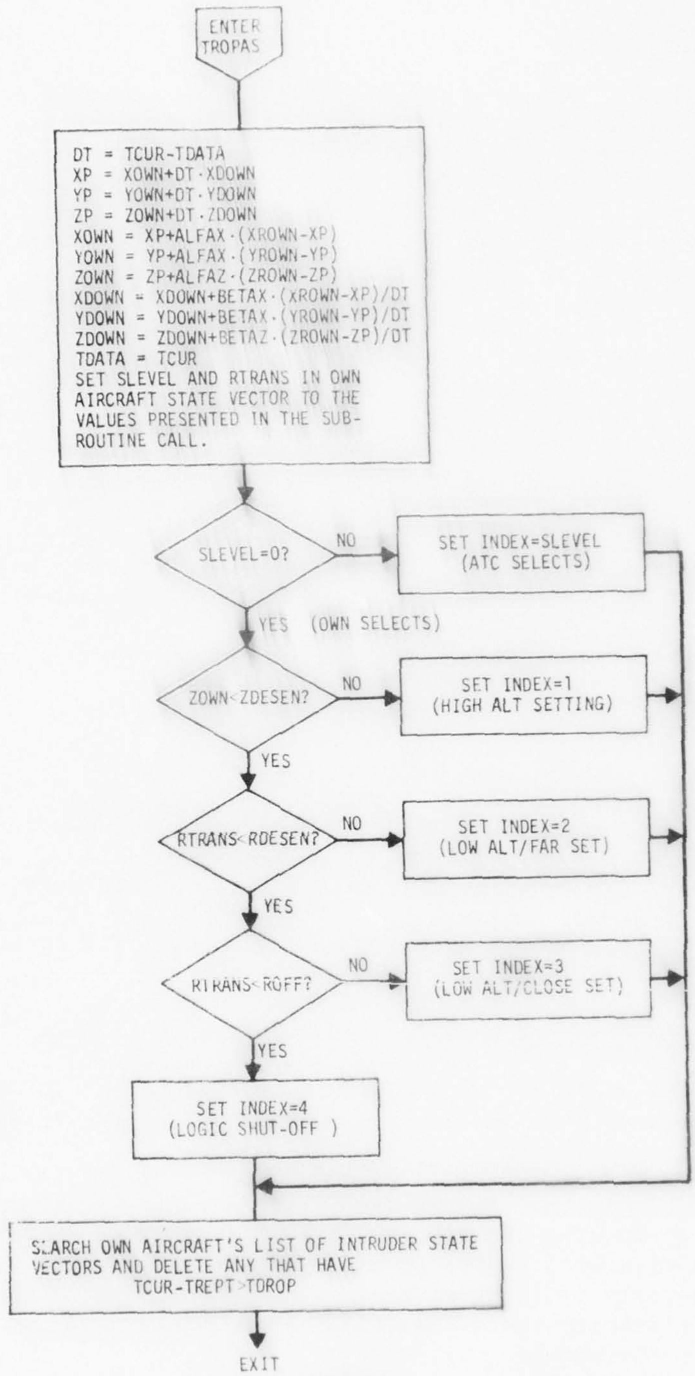


FIGURE 7-1
TRACKING OWN DATA FOR THE PASSIVE MODE (TROPAS)

TABLE 7-1

ARGUMENTS IN CALL TO TROPAS

SYMBOL	MEANING
TCUR	Current time
IDOWN	Identification of aircraft to receive own tracking
XROWN	Own aircraft's X coordinate report
YROWN	Own aircraft's Y coordinate report
ZROWN	Own aircraft's Z coordinate report
SLEVEL	Own aircraft's sensitization level as directed by the ground system
RTRANS	Own aircraft's range from a fixed ground transponder that is used for desensitization

ALFAX, ALFAZ, BETAX, and BETAZ are parameters and are listed in Appendix B. TDATA is a variable in own aircraft's state vector which gives the time for which the coordinates in own aircraft's state vector are represented. Other names are either local variables or variables with obvious meaning in own aircraft's state vector.

This subroutine also contains logic to set the desensitization level of the BCAS logic. Since the desensitization level is a function of own aircraft's location, this can be done once per BCAS logic cycle in this routine. The result is contained in the variable INDEX, which is stored in own aircraft's state vector for use during the remainder of the BCAS cycle. The level of desensitization can be set by the ground ATC system through data link or it can be determined by the BCAS logic itself.

In the first case, ATC can select one of four possible levels by setting the input variable SLEVEL to 1, 2, 3 or 4. If ATC sets SLEVEL to 4, the collision avoidance logic is shut-off and the tracking program is not executed. In normal operation, however, only the first three levels are selected. The first level (SLEVEL=1) is the non-desensitized setting. It is normally selected when own aircraft is at high altitude. The second level (SLEVEL=2) is the partially desensitized setting, and it is normally selected when own aircraft is at low altitude but far out. Finally, the third level (SLEVEL=3) is normally considered the fully desensitized setting and is selected when own aircraft is at low altitude and close in. As indicated, the ground can select any one of these desensitization levels by setting SLEVEL to the proper <u>positive</u> integer.

However, as shown, if SLEVEL=0, the program itself selects one of the first three desensitization levels or shuts off the collision avoidance logic. The logic bases its selection on own altitude, ZOWN, and distance, RTRANS, from a ground-based transponder, choosing one of the following:

1. the non-desensitized level (i.e., high altitude setting) <u>if</u> ZOWN $\geq$ ZDESEN,

2. the partially desensitized level (i.e., low altitude but far out setting) <u>if</u> ZOWN < ZDESEN and RTRANS $\geq$ RDESEN,

3. the fully desensitized level (i.e., low altitude and close in setting) <u>if</u> ZOWN < ZDESEN and ROFF < RTRANS < RDESEN,

or 4. the collision avoidance logic is shut-off <u>if</u> ZOWN < ZDESEN and RTRANS $\leq$ ROFF

If one of the first 3 levels is selected, either via ground control or program selection, desensitization is accomplished by properly initializing certain critical parameters affecting desensitization. Each parameter is set to one of six possible values according to the desired desensitization level and the intruder's equipage.

The following table summarizes the five integer values that can be assumed by the input variable SLEVEL:

SLEVEL VALUE	MEANING
0	Own aircraft selects desensitization level
1	ATC picks non-desensitized level (high altitude setting)
2	ATC picks partial desensitization (low altitude/far setting)
3	ATC picks full desensitization (low altitude/close setting)
4	ATC shuts-off collision avoidance logic

7.2 Tracking Own Data for the Active Mode (TROACT)

The flow chart for this subroutine is presented in Figure 7-2 and the arguments in the subroutine call are given in Table 7-2. The subroutine is analogous to TROPAS except that tracking is conducted on range instead of on X and Y coordinates.

7.3 Tracking Intruder Data for the Passive Mode (TRIPAS)

The flow chart for this subroutine is presented in Figure 7-3 and the arguments in the subroutine call are given in Table 7-3. This flow chart contains logic to create or eliminate a state vector for the given intruder. The intruder is dropped if a period of time equal to TDROP has passed without a target report. The variable in the intruder state vector, TREPT, which is the time of the last good reports, is maintained to permit this determination. For real-world implementation, reasonability tests would be applied to all reports before they would be

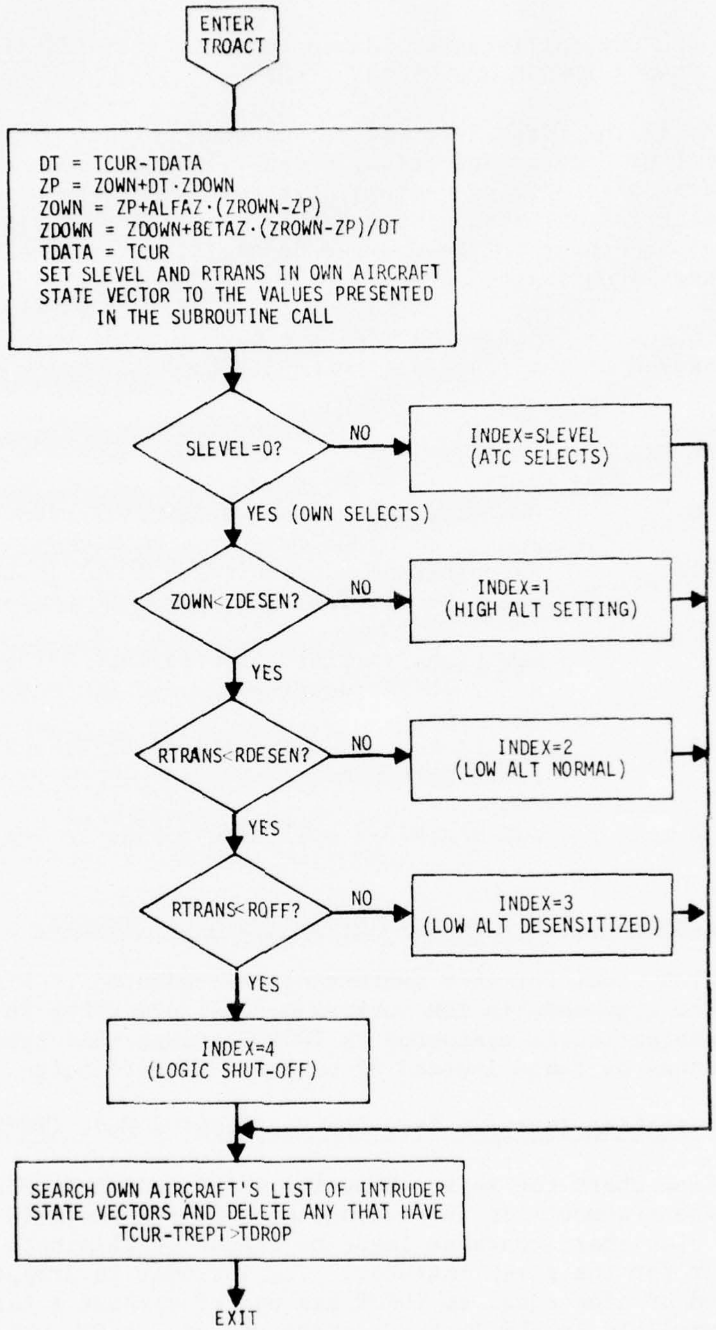


FIGURE 7-2
TRACKING OWN AIRCRAFT DATA FOR THE ACTIVE MODE (TROACT)

TABLE 7-2

ARGUMENTS IN CALL TO TROACT

SYMBOL	MEANING
TCUR	Current time
IDOWN	Identification of aircraft to receive own tracking
ZROWN	Own aircraft's Z coordinate report
SLEVEL	Own aircraft's sensitization level as directed by the ground system
RTRANS	Own aircraft's range from a fixed ground transponder that is used for desensitization

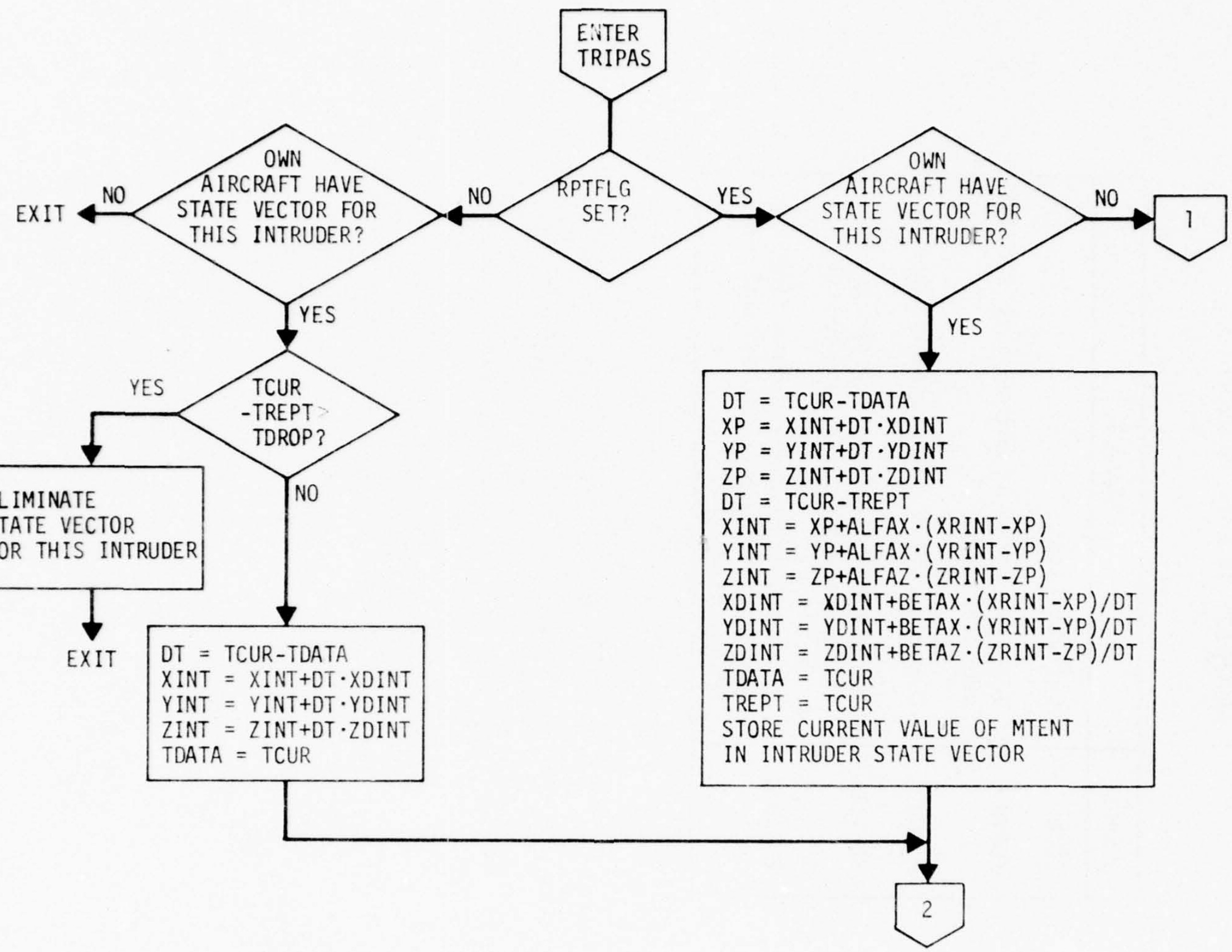


FIGURE 7-3
TRACKING INTRUDER DATA FOR PASSIVE MODE (TRIPAS)

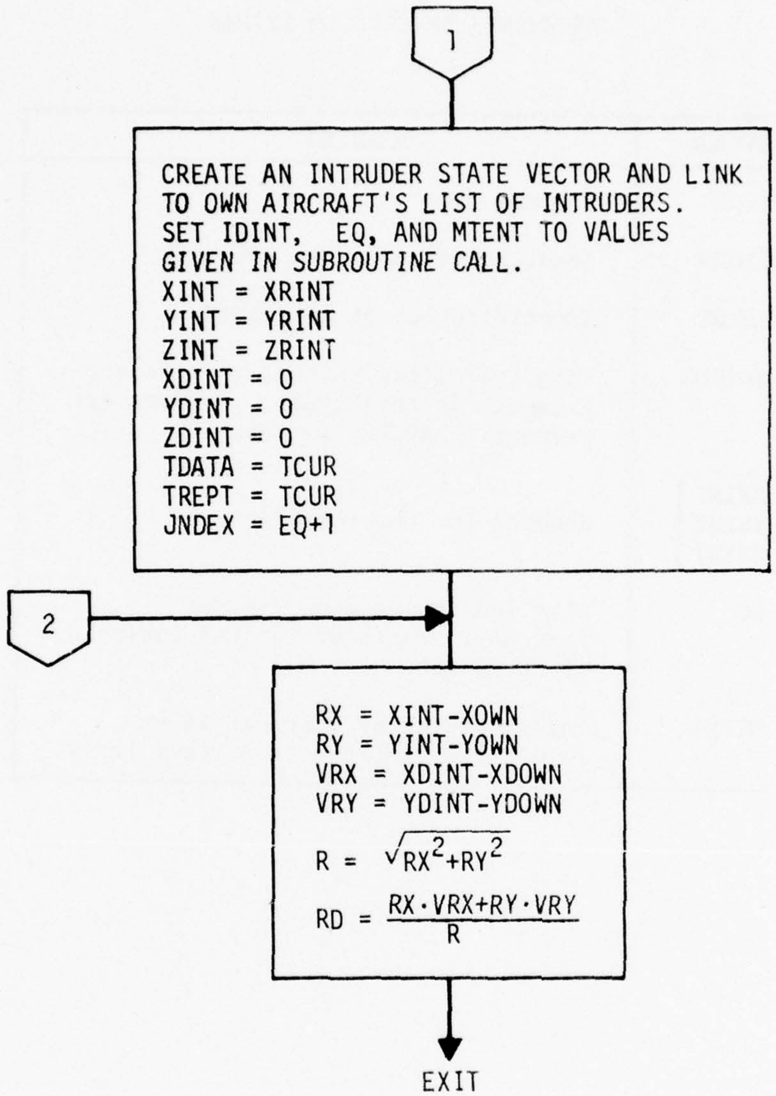


FIGURE 7-3 (Concluded)
TRACKING INTRUDER DATA FOR PASSIVE MODE (TRIPAS)

TABLE 7-3

ARGUMENTS IN CALL TO TRIPAS

SYMBOL	MEANING
TCUR	Current time
IDOWN	Identification of own aircraft
IDINT	Identification of intruder
RPTFLG	Flag indicating whether reports are present for this cycle. Reports are present if RPTFLG is set.
XRINT YRINT ZRINT	Reports for the intruder
EQ	Flag indicating equippage of intruder. Intruder is BCAS equipped if EQ is set.
MTENT	Intent status of intruder is obtained from intruder's current reply.

accepted for updating the track. The simulation environment is not simulating unreasonable reports, so these tests are omitted here.

The flow chart includes logic to coast the intruder's track when a missed report is simulated. All position coordinates are extrapolated linearly and the velocity coordinates are left unchanged. The structure of the logic specified in this document requires that the external program call the BCAS logic each BCAS logic cycle whether or not there is a missed report. Whether a report is present or not is indicated by the flag RPTFLG.

Note that the time difference, TCUR-TDATA, is used for predicting the position coordinates for the current scan but the difference TCUR-TREPT is used in the velocity tracking equations.

Since all coordinates are maintained and tracked in X and Y in the passive mode, it is necessary to compute R and RD after tracking.

When a new intruder first appears, a new state vector is created and linked to the list of intruders for own aircraft. The track is started by using the reports directly for the position coordinates and by starting all velocity coordinates at zero. Within 6 or 7 BCAS cycles, the tracked coordinates will have converged to within reasonable limits of the true coordinates.

7.4 Tracking Intruder Data for the Active Mode (TRIACT)

The flow chart for this subroutine is presented in Figure 7-4, and the arguments in the subroutine call are given in Table 7-4. The flow chart is fully analogous to TRIPAS.

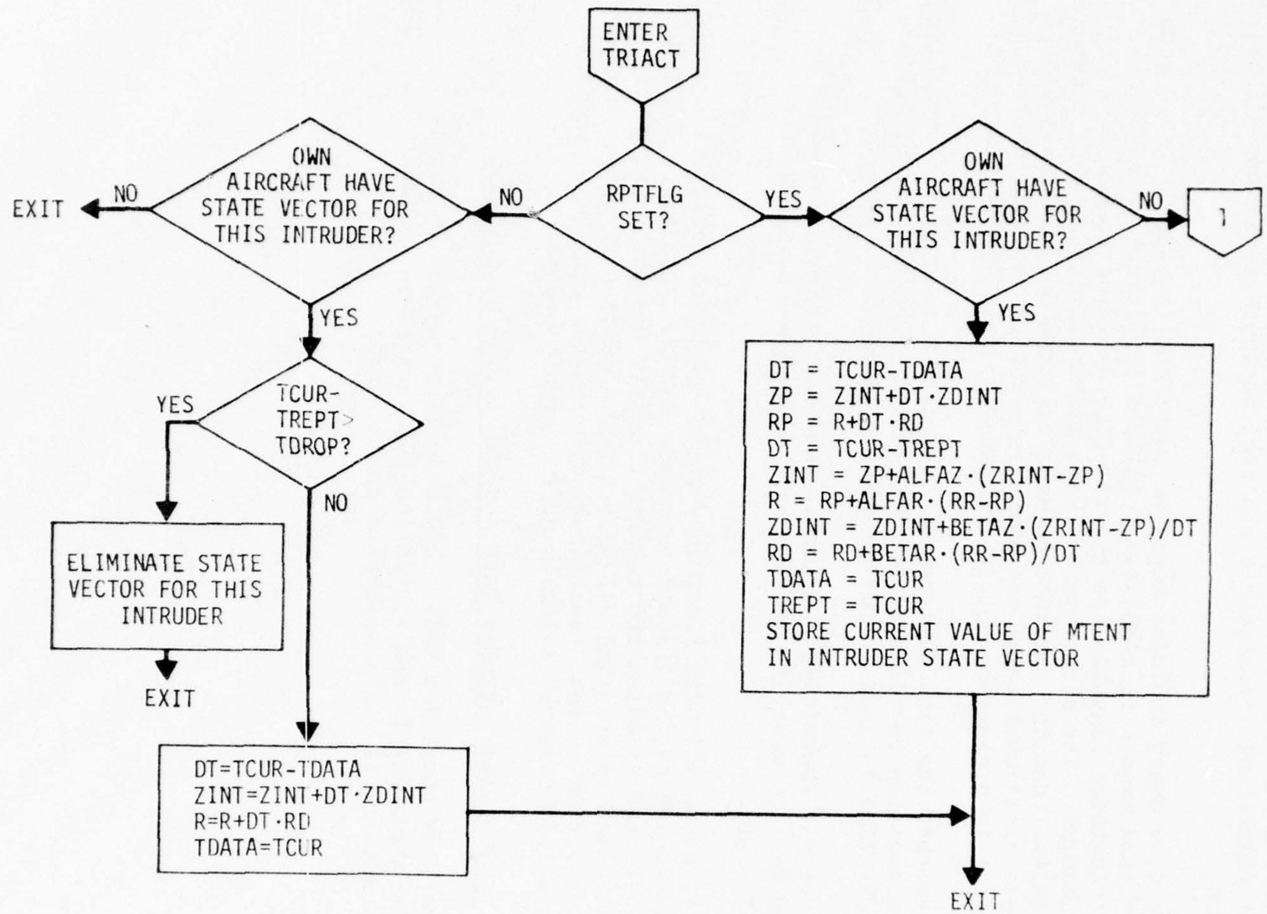


FIGURE 7-4
TRACKING INTRUDER DATA FOR ACTIVE MODE (TRIACT)

1

CREATE AN INTRUDER STATE VECTOR AND LINK TO OWN AIRCRAFT'S LIST OF INTRUDERS. SET IDINT, EQ, AND MTENT TO VALUES GIVEN IN THE SUBROUTINE CALL.
ZINT = ZRINT
R = RR
ZDINT = 0
RD = 0
TDATA = TCUR
TREPT = TCUR
JNDEX = EQ+1

EXIT

FIGURE 7-4 (Concluded)
TRACKING INTRUDER DATA FOR ACTIVE MODE (TRIACT)

TABLE 7-4

ARGUMENTS IN CALL TO TRIACT

SYMBOL	MEANING
TCUR	Current time
IDOWN	Identification of own aircraft
IDINT	Identification of intruder
RPTFLG	Flag indicating whether reports are present for this cycle. Reports are present if RPTFLG is set.
ZRINT	Z report of the intruder
RR	Range report for the intruder
EQ	Flag indicating equipage of intruder. Intruder is BCAS equipped if EQ is set.
MTENT	Intent status of intruder as obtained from intruder's current reply.

APPENDIX A

TESTING OWN INTENT AND INTRUDER'S INTENT FOR COMPATIBILITY

To test whether own intent code NTENT is compatible with the intruder's intent MTENT, the program compares the absolute values $|NTENT|$ and $|MTENT|$. The following matrix shows which codes are compatible (C) and which are incompatible (I).

		$\lvert MTENT \rvert$						
		0	1	2	3	4	5	6
$\lvert NTENT \rvert$	0	C	C	C	C	C	C	C
	1	C	I	C	C	C	C	C
	2	C	C	I	C	C	C	C
	3	C	C	C	I	I	I	C
	4	C	C	C	I	C	I	I
	5	C	C	C	I	I	C	I
	6	C	C	C	C	I	I	I

If the matrix resides in memory, comparing the two codes for compatibility is very easy, but core limitations may make other methods of comparison more desirable. The matrix shows that a vertical code (i.e., 1 or 2) is always compatible with a horizontal code (i.e., 3, 4, 5 or 6). Hence, two codes may be incompatible only if they are both horizontal or both vertical, in which case they are compatible only if $|NTENT| = ICOMP(|MTENT|)$ where the array ICOMP is defined in Section 3.

APPENDIX B

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS (ALPHABETICAL ORDER)

Table B-1 identifies system parameters of the logic described in this document and briefly describes their utilization. Nominal values are given to assist understanding the logics. Most of the parameters are used in both the passive and the active logics, but a few are special to only one logic. If a parameter is used in the passive logic, an "X" will appear opposite the parameter in column P. Similarly, if a parameter occurs in the active logic, an "X" will appear in column A. There are 6 nominal values for each parameter affecting desensitization: the first 3 values apply to the unequipped case, and the second 3 apply to the equipped case; they are stored in order of increasing desensitization within each grouping of 3.

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
ACCEPT	Altitude separation within which own aircraft accepts intruder's slection of vertical command when own selection is incompatible.	X	X	400 ft.
ALFAR	Tracking constant for range		X	0.4
ALFAX	Tracking constant for X and Y position	X		0.4
ALFAZ	Tracking constant for Z position	X	X	0.4
ALIM1	Vertical miss distance within which positive rather than negative vertical commands are requested.		X	700 ft. 400 400 700 400 400
ALIM2	Vertical miss distance in excess of which negative vertical commands are requested rather than horizontal commands or positive vertical commands	X		700 ft. 400 400 700 400 400
BAND1	Altitude separation within which vertical rate is limited to a maximum of 500 ft./minute	X	X	1300 ft. 1000 800 1300 1000 800

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Continued)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
BAND2	Altitude separation within which vertical rate is limited to a maximum of 1000 ft./min.	X	X	1800 ft. 1500 1000 1800 1500 1000
BETAR	Tracking constant for range rate		X	0.15 ft.
BETAX	Tracking constant for X and Y velocity	X		0.15 ft.
BETAZ	Tracking constant for Z velocity	X	X	0.15 ft.
DMOD	Modified-tau distance used for positive and negative commands	X	X	1.8 nmi 0.75 0.3 1.8 0.75 0.3
DMODP	Modified-tau distance used for IPD detection	X		1.8 nmi 0.75 0.50 1.8 0.75 0.50

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Continued)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
LALT	Altitude separation outside which vertical rate limit commands are not given	X	X	3300 ft. 1500 1000 3300 1500 1000
MDCMD	Square of horizontal miss distance threshold beyond which no positive or negative maneuvers are requested by threat detection logic	X		9.0 nmi^2 4.0 1.0 9.0 4.0 1.0
MDIPDF	Square of horizontal miss distance threshold used in flashing IPD test	X		1.0 nmi^2 0.25 0.25 1.0 0.25 0.25
MDPOS	Square of horizontal miss distance threshold used by threat detector to choose between positive and negative command requests	X		1.0 nmi^2 0.25 0.16 1.0 0.25 0.16

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Continued)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
MTAU2	Modified-tau distance used to determine whether vertical limit commands should be given	X	X	1.8 nmi 0.75 0.3 1.8 0.75 0.3
PFUN	Weights used to estimate the predicted vertical miss distance	X	X	4.0 sec. 7.2 9.8 11.8 13.5
RDESEN	Range threshold used to desensitize logic at low altitude	X	X	15 nmi
RDTHR	Range-rate threshold use to choose between Tau test and immediate range test	X	X	10 ft./sec.
ROFF	Range threshold used by program to shut off collision avoidance logic	X	X	2 nmi
RTHPF	Immediate range threshold for flashing IPD tests	X		1.0 nmi 0.5 0.3 0.5 0.3 0.3

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Continued)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
RTHPO	Immediate range threshold for ordinary IPD tests	X		3.0 nmi 2.0 1.0 3.0 2.0 1.0
RTHR	Immediate range threshold used in threat detection for immediate range test	X	X	2.0 nmi 0.75 0.3 1.0 0.5 0.3
RZIPDF	Immediate altitude threshold for flashing IPD tests	X		700 ft. 400 400 700 400 400
RZIPDO	Immediate altitude threshold for ordinary IPD tests	X		2000 ft. 1500 1000 2000 1500 1000

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Continued)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
SMALL1	Small value used to avoid dividing by zero when computing TAU2.	X	X	10 kts.
TAU2L	Threshold against which TAU2 is compared to determine whether limit commands should be given	X	X	40 sec. 40 40 40 40 40
TDROP	Time without reported data to drop an intruder	X	X	10 sec.
TIMETX	In horizontal resolution, the time to track crossing point threshold	X		10 sec.
TIMEV	Look-ahead time used to compute the projected vertical miss distance VMD to determine whether to request a positive command or a negative command		X	25 sec. 20 20 25 20 20
TIPDF	Tau threshold for flashing IPD tests	X		35 sec. 30 30 35 30 30

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Continued)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
TIPDO	Tau threshold for ordinary IPD tests	X		60 sec. 60 60 60 60 60
TLARGE	Very large positive number	X	X	10^5 sec.
TPOMIN	Minimum time positive command can be displayed	X	X	5 sec.
TRTHR	Value against which modified-tau (TAUR) is being compared	X	X	30 sec. 30 25 25 25 20
TVPCMD	Look-ahead time used to compute the projected vertical miss distance VMD	X	X	25 20 20 25 20 20

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Continued)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
TVTHR	Value against which vertical tau (TAUV) is being compared	X	X	30 sec. 30 25 25 25 20
TV1	Look-ahead time used to choose climb or descend command	X	X	8 sec.
TXTH	In horizontal resolution algorithm, the track crossing angle at which the resolution strategy changes	X		90°
WAITM	Maximum reply wait time.	X	X	4.0 sec.
ZDESEN	Altitude threshold below which logic is desensitized	X	X	10,000 ft.
ZDTHR	Altitude rate threshold used by the threat detection (Note: ZDTHR = -ZTHR/TVTHR)	X	X	-30 ft./sec. -30 -36 -36 -36 -45
ZIPD	Altitude threshold for co-altitude IPD	X		500 ft.

TABLE B-1

BCAS COLLISION AVOIDANCE LOGIC PARAMETERS
(Concluded)

SYMBOL	UTILIZATION	P	A	NOMINAL VALUE
ZTHR	Immediate altitude threshold used by detection logic	X	X	900 ft. 900 900 900 900 900

APPENDIX C

DISPLAY INDICATOR CODES

DPLY	COMMAND DISPLAYED
0	no command selected
1	don't climb
2	don't descend
3	level off
4	(not used)
5	climb
6	descend
7	don't climb faster than 500 ft./min.
8	don't climb faster than 1000 ft./min.
9	don't climb faster than 2000 ft./min.
10	don't descend faster than 500 ft./min.
11	don't descend faster than 1000 ft./min.
12	don't descend faster than 2000 ft./min.
13	don't turn right
14	don't turn left
15	turn left
16	turn right

APPENDIX D

THE FUNCTION P(T) USED TO COMPUTE THE VERTICAL MISS DISTANCE VMD WHEN $\dot{A} > 0$ AND INDEX = 3

The function P(T) is evaluated at time T = TAUR in order to compute the projected vertical miss distance VMD in TAUR seconds from the current time. The distance VMD is set to $A + \dot{A}*P(TAUR)$ and then compared to the threshold ALIM to determine whether a positive vertical maneuver should be given.

P(0) = 0 since the current vertical separation is A. Furthermore, since $\dot{A} > 0$, P(T) is an increasing function of time. However, to be on the safe side in projecting the vertical separation and deciding whether to request a positive command, P(T) is assumed to be less than the linear projection, i.e., P(T) < T; and to reflect the increasing uncertainty of $\dot{A}$ as T increases, the rate of growth of P(T) is assumed to decrease in time. There are many functions that have the foregoing desired shape, for example, the area under the curve e^{-aT} in the interval $0 \leq T \leq$ TRTHR where the constant a is chosen to reflect the increasing uncertainty of $\dot{A}$.

For the purpose of the NAFEC simulation, P(T) is approximated by a piece-wise linear function whose linear segments have endpoints at T = 0, 5, 10, 15, 20, and 25 seconds. The values of P at these end-points are stored in the array PFUN (see Appendix B).* Approximating P in this manner permits very easy programming. Figure D-1 shows the piece-wise linear approximation. Note that P(T) = P(TRTHR) for T > TRTHR.

---

* P(0) = 0 is not stored.

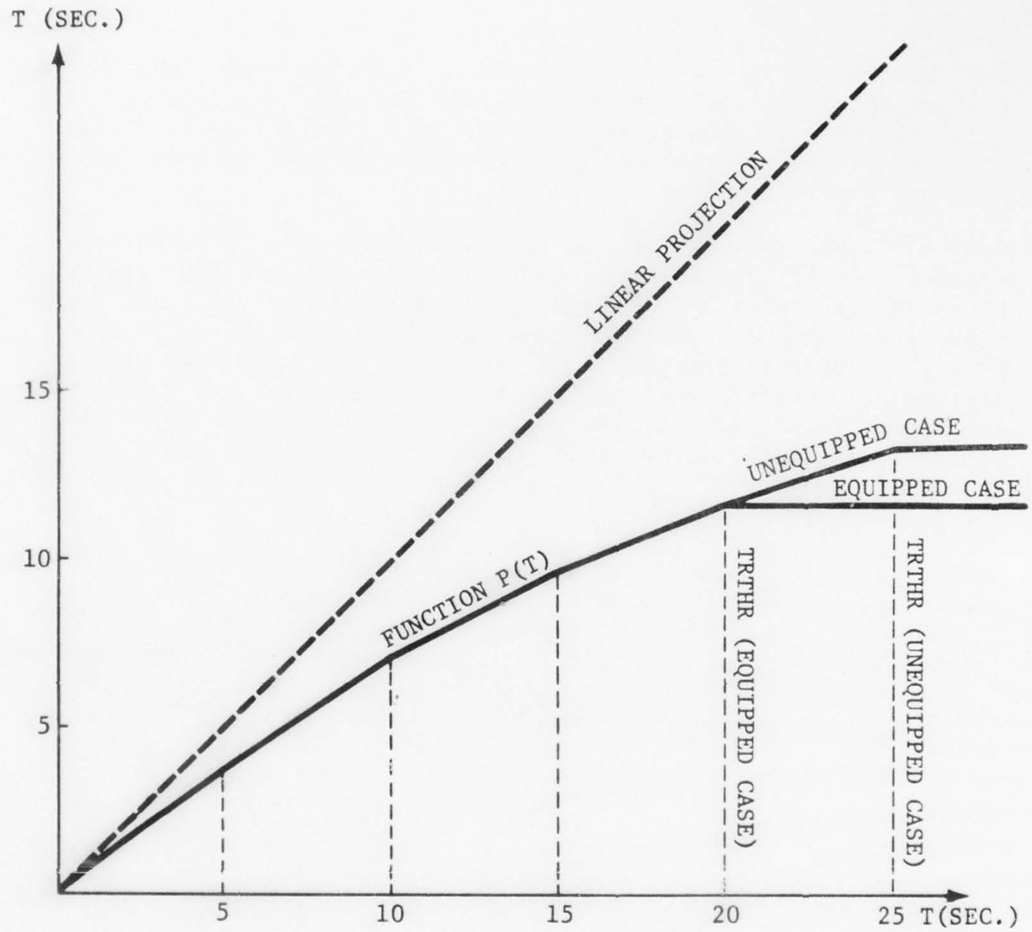


FIGURE D-1
FUNCTION P(T) FOR BOTH EQUIPPED AND UNEQUIPPED CASES

AD-A047 254 MITRE CORP MCLEAN VA METREK DIV F/G 17/7
INITIAL COLLISION AVOIDANCE ALGORITHMS FOR THE BEACON-BASED COL--ETC(U)
APR 77 J CLARK, A MCFARLAND DOT-FA69NS-162
UNCLASSIFIED MTR-7532 FAA-RD-77-163 NL

2 OF 2
AD A047254

END
DATE
FILMED
1 -78
DDC

APPENDIX E

REFERENCES

1. Air Transport Association of America, "Airborne Collision Avoidance System," ANTC Report No. 117 (Rev. 10), May 1971.

2. A. L. McFarland, B. M. Horowitz, "A Description of the Intermittent Positive Control Concept," The MITRE Corporation, McLean, Virginia, FAA-EM-74-1, (Rev. 1), July 1975.

GPO 925 036